NONCONVEX NETWORK OPTIMISATION:
SOLUTION METHODS AND APPLICATIONS

A THESIS

SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENTS FOR THE DEGREE

OF

DOCTOR OF PHILOSOPHY IN MANAGEMENT SCIENCE

IN THE

UNIVERSITY OF CANTERBURY

by

Gavin J. Bell

University of Canterbury

1998

Acknowledgements

The production of this thesis would not have been possible were it not for the advice, insight, and support given by my supervisors, colleagues, and friends. Firstly, thanks to Bruce Lamar and Grant Read for their guidance, encouragement, and supervision. I particularly want to thank Bruce for his efforts since resigning from the University of Canterbury. Continuing to supervise a thesis whilst holding a non-academic position in another country is not an easy task.

I also wish to acknowledge the support, both financial and otherwise, given to me by the Energy Modelling Research Group (EMRG) during the period of this research, and to thank the staff and students involved with EMRG for their help and assistance. In particular, I would like to thank Chris Wallace for his work on NetSpeak. Thanks are also due to the other academic and secretarial staff of the Department of Management.

Finally, thank you to my friends and family for their support. My greatest thanks goes to Rachel, for her continual support, friendship and love.

Abstract

Nonconvex network flow models are used in a wide variety of problem domains involving discounting or economies of scale. Examples include network design, facility location, production planning, physical distribution, electricity transmission, and telecommunications problems. A variety of methods for solving nonconvex network flow problems have been developed in the literature; however, global optimisation of this class of problem is complex, and the convergence of the procedures slow. In this thesis we develop techniques that can be used to enhance standard solution procedures for nonconvex network flow problems. A general theory of concave underestimator analysis for such problems is presented. Based on this work, the theory of enhanced capacity improvement is developed and presented as part of a branch and bound solution algorithm for minimum cost mixed-integer nonconvex network flow problems with side constraints. Computational analysis of this algorithm is described which shows the branch and bound algorithm incorporating enhanced capacity improvement provides a substantial performance increase over the same algorithm without capacity improvement. Finally we use and extend capacity improvement to develop an algorithm for the problem of short term electricity dispatch.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern society is replete with a multitude of network–like systems. Important examples include the generation and transmission of electricity; transportation and physical distribution of goods and services; analysis, operation and design of computer networks and telecommunications systems; location of facilities such as depots and factories; and waste management systems to name a few. Often, such systems have a design or structure that falls naturally into the flexible and robust modelling framework provided by network flow models. Consequently, the field of network optimisation, including the analysis and development of network flow models and solution techniques, is one of the most important and useful in operations research.

As well as accurately representing many "real–world" problem situations and systems, network models are also characterised by an easily constructed and understood diagrammatical or pictorial representation. Such a feature has several advantages. First, and most importantly, the symbolic nature of network models greatly enhances the communication of the ideas underlying the model and its analysis to people with little or no background in network optimisation. Increasing the comprehensibility of a model increases its credibility, and therefore the prospects of any resulting analysis and insights being accepted and implemented. Second, the mathematical description of the network model can be easily obtained from the network diagram. This feature aids the process of model development – the physical structure of a network easily lends itself to diagrammatic representation, which can then be transformed to a mathematical model formulation in a straightforward manner. It has even been conjectured, in Glover, Klingman & Phillips (1992), that the combination of the pictorial and mathematical aspects of network models assists the coordination of the right and left hemispheres of the brain leading, to a greater facility in problem solving.

1.1 What is a network?

A network is essentially a group of points, or *nodes*, that are connected together by a series of links called *arcs*. For example, an electricity transmission system can be viewed as a network, in which electricity is transmitted from generation points to towns and cities (the nodes) via transmission lines (the arcs). Another example is the distribution network of a logging company, in which logs are harvested and transported to processing plants before being transported to customers (for example lumber retailing stores). Here the nodes are the harvesting locations (i.e. forests), the company's processing plants, and the company's customers. The arcs correspond to the transportation system linking the forests, plants, and customers together.

Networks have an associated *objective* or "aim", determined by the decision maker. For example, the objective of a network that models the transportation of goods from factories to customers may be to minimise the cost of the transportation schedule. A second example is a network that models the production process at a factory. The objective may then be to maximise profit from production. The objective of a network is expressed as a mathematical *function* of the "activity" of the network. For example, in the transportation network example, the objective is to minimise the cost of transporting goods from factories to customers. The *objective function* of the network will be a mathematical function of the flow of goods in the network, giving the total cost of any transportation schedule. The "aim" will then be to find that transportation schedule that minimises the value of the objective function.

Each node in a network may have an associated *supply* and a *demand*. A supply is simply an input into the network, and a demand an output from the network. For example, in an electricity transmission network, electricity is supplied to the network at the generator nodes, and removed from the network ("demanded") in the towns and cities. In the lumber distribution network example, lumber is supplied at the nodes corresponding to the forests, and removed from the network at the nodes corresponding to the company's customers. Nodes that originate flow only (that is, nodes that have a supply and no incoming arcs) are called *supply* or *source* nodes. Nodes that receive or terminate flow only (that is, nodes that have a demand and no out-going arcs) are called *demand* or *sink* nodes. Nodes that both receive and send flow (and may have an associated net supply or demand) are called *transshipment* nodes.

Each arc in a network is associated with a decision in the system modelled by the network. In the electricity transmission network, the decisions to be made are the amount of electrical flow to send over each network arc. That is, the decision to be made for each transmission line, or arc, is how much electricity to send on it. The "value" or "level" of the decision on a particular arc is called the *flow* on that arc. For example, the flow on a transmission line may be set to 100 MW. In the lumber distribution network, one decision may be the amount of flow, or lumber, to transport on the arc from, say, a particular timber mill to a particular hardware store.

The flow on each arc has a minimum and a maximum allowable level. These arc flow bounds or limits can have a variety of causes, such as contractual agreements or physical limits. For example, in the electricity transmission network a transmission line may have a maximum allowable level of electrical flow, beyond which damage to the line might occur. In the lumber distribution network, there may be a Union agreement which specifies a minimum number of logs that have to be shipped from forest A to plant D.

The cost on an arc is a measure of the effectiveness of the network's decisions in terms of the decision maker's objective(s). It may be a monetary cost – for example the marginal, or per–unit, cost of transporting a good from a factory to a warehouse. For example, a company may be charged $10 per unit to transport goods from point A to point B. Therefore, the cost associated with the arc from point A to point B in the company's transportation network model would be \$10 per unit. A flow of 15 units on that arc would cost a total of \$150. Alternatively, it may be some other notional or physical aspect of the system. For example, a network may be constructed to assign students (each student would be correspond to a demand node) to one of a number of possible tutorial classes (each tutorial class would correspond to a supply node). The students may be asked to rank the tutorials in order of preference. The cost on the arc linking student A to tutorial B would then be the ranking given to that tutorial by the student. The costs are linked directly to the objective of the network. In the transportation network example, the objective could be to minimise total transportation costs or transportation time. In the student–tutorial assignment network, the objective could be to maximise the satisfaction (or, equivalently, minimise the dissatisfaction) of the student body as a whole with their class assignments.

To illustrate the type of nonconvex network optimisation methods considered in this thesis, we first consider an example transportation network based on one presented in Daellenbach, George & McNickle (1983). A New Zealand carpet manufacturer produces the same type of carpet in two factories. The first factory, located in Christchurch, produces 55 rolls of carpet per week. The second factory, located in Auckland, produces 50 rolls. The carpet is sold via five regional distribution warehouses. These are located in Auckland, Rotorua, Wellington, Christchurch, and Dunedin. The weekly demand for carpet in each of the five warehouses is 30, 10, 25, 20, and 20 rolls respectively. Carpets can be transported to each of the distribution warehouses either directly from the factories or via the large Wellington

Table 1.1: Linear transportation costs for carpet distribution network

Arc (*from - to*)	Cost ($ per Unit)
Auckland - Auckland	7.00
Auckland - Rotorua	14.00
Auckland - Wellington	18.00
Auckland - Christchurch	30.00
Auckland - Dunedin	34.00
Christchurch - Auckland	30.00
Christchurch - Rotorua	24.00
Christchurch - Wellington	20.00
Christchurch - Christchurch	5.00
Christchurch - Dunedin	15.00
Wellington - Auckland	18.00
Wellington - Rotorua	15.00
Wellington - Christchurch	20.00
Wellington - Dunedin	25.00

warehouse. Table 1.1 gives the cost per roll to transport the carpet from each factory to each warehouse. The problem is to determine a transportation schedule, from the two factories to the five warehouses, that has the lowest possible cost. The physical transportation network and its diagrammatical network representation are given in Figures 1.1 and 1.2 respectively.

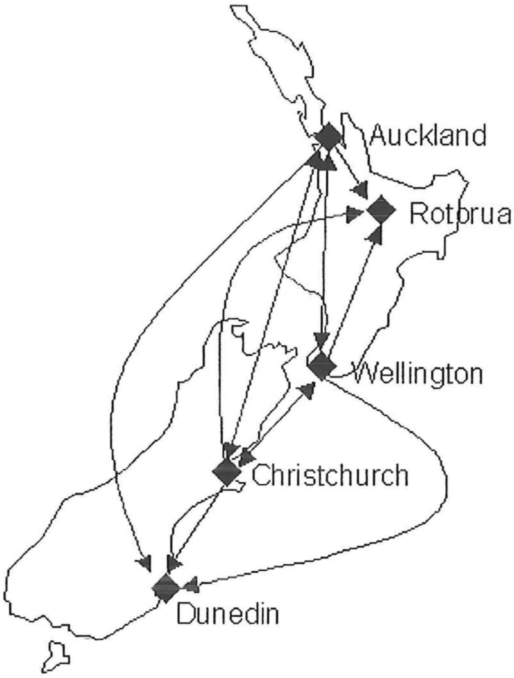

Figure 1.1: Carpet distribution network with linear costs

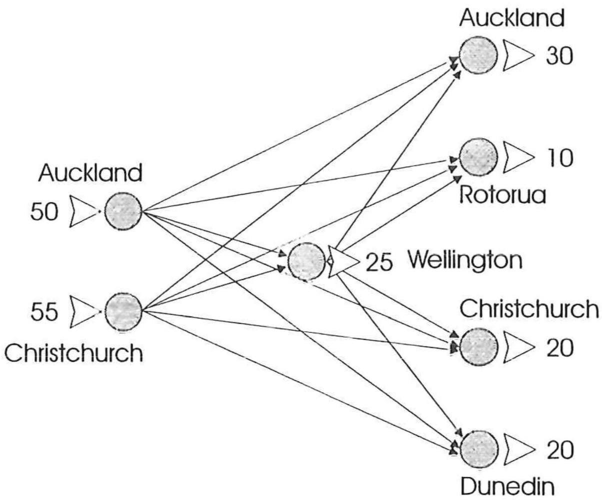

Figure 1.2: Network Diagram of linear carpet distribution network

1.1.1 Mathematical Representation of Pure Network Flow Models

One major benefit of network flow models is that the mathematical programming description of a network can obtained directly from the network diagram. We first represent the flow on generic arc j with an arc flow variable x_j. The flow, or decision, variables can have lower and upper bounds associated with them, denoted l_j and u_j respectively; that is $l_j \leq x_j \leq u_j$. The objective of the network (for example, to minimise costs) is a function of the arc flow variables. In traditional, pure network flow models, the objective function is an affine (linear) function of the flow variables. Finally, in pure network flow models the flow into a node, from both supply to that node and arcs into that node, must equal the flow leaving that node, from both demand at the node and arcs leaving the node. These "conservation of flow" requirements are modelled as problem constraints.

Thus the generic mathematical programming formulation for pure network flow models is as follows:

$$\begin{aligned} \min \quad & \underline{c}^{\mathrm{T}} \cdot \underline{x} = \textstyle\sum_{j \in J} c_j \cdot x_j & \\ \text{s.t.} \quad & \textstyle\sum_{j \in L_i} x_j - \sum_{j \in E_i} x_j = b_i & \forall i \in I \\ & l_j \leq x_j \leq u_j & \forall j \in J \end{aligned} \tag{1.1}$$

where I is the node set with generic element i and cardinality m, J is the (directed) arc set with generic element j and cardinality n, F_j (T_j) is the "from" ("to") node for arc j, $L_i = \{j : F_j = i\}$ is the set of arcs "leaving" node i, $E_i = \{j : T_j = i\}$ is the set of arcs "entering" node i, b_i is the supply (if $b_i > 0$) or demand (if $b_i < 0$) at node i, $\underline{x} = (\ldots, x_j, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the arc flow decision variable vector, $\underline{l} = (\ldots, l_j, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the flow lower bound vector, $\underline{u} = (\ldots, u_j, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the flow upper bound vector, and $\underline{c} = (\ldots, c_j, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the vector of marginal, or per–unit, costs for the arcs $j \in J$.

It can be convenient to re-express the network formulation above using a more compact notation. We define the matrix A, called the *node-arc incidence matrix*, as the matrix of network conservation of flow coefficients. That is, a_{ij}, the element of A in the i–th row and j–th column, equals 1 if $j \in L_i$, equals -1 if $j \in E_i$, and equals 0 otherwise. We then define the polyhedron $X \in \mathbb{R}^n$ as the set of arc flow variable values satisfying the conservation of flow constraints; that is $X = \{\underline{x} : A\underline{x} = \underline{b}\}$. We also define $H = \{\underline{x} : \underline{l} \leq \underline{x} \leq \underline{u}\} \in \mathbb{R}^n$ as the hyperrectangle defined by the lower and upper flow bounds on the flow variables. The pure network formulation expressed above becomes

$$\min \quad \underline{c}^{\mathrm{T}} \cdot \underline{x} \text{ s.t. } \underline{x} \in S = X \cap H$$

The formulation for the example transportation problem described previously (see Table 1.1 and Figures 1.1 and 1.2) is as follows:

$$
\begin{array}{lll}
\min & 7x_{A-A} + 14x_{A-R} + 18x_{A-W} + 30x_{A-C} + 34x_{A-D} & \\
 & 30x_{C-A} + 24x_{C-R} + 20x_{A-W} + 5x_{C-C} + 15x_{C-D} & \\
 & 18x_{W-A} + 15x_{W-R} + 20x_{W-C} + 25x_{W-D} & \\
 & x_{A-A} + x_{A-R} + x_{A-W} + x_{A-C} + x_{A-D} = 50 & \text{(FA)} \\
 & x_{C-A} + x_{C-R} + x_{C-W} + x_{C-C} + x_{C-D} = 55 & \text{(FC)} \\
 & -x_{A-A} - x_{W-A} - x_{C-A} = -30 & \text{(HA)} \\
\text{s.t.} & -x_{A-R} - x_{W-R} - x_{C-R} = -10 & \text{(HR)} \\
 & -x_{A-W} - x_{C-W} + x_{W-A} + x_{W-R} + x_{W-C} + x_{W-D} = -25 & \text{(HW)} \\
 & -x_{A-C} - x_{W-C} - x_{C-C} = -20 & \text{(HC)} \\
 & -x_{A-D} - x_{W-D} - x_{C-D} = -20 & \text{(HD)}
\end{array}
$$

where each city is referenced by its first letter, x_{X-Y} is the decision (flow) variable corresponding to the arc from X to Y, (F*) labels the constraint corresponding to factory *, and (H*) labels the constraint corresponding to warehouse *.

Solving this as a pure network flow problem yields the solution presented in Figure 1.3. The optimal solution is to ship all carpets directly to the warehouses from the factories, rather than through the "super-warehouse" in Wellington. This solution is intuitively appealing, since the costs were (roughly) proportional to the distances between cities.

From	To	Flow
Auckland	Auckland	30
Auckland	Rotorua	10
Auckland	Wellington	10
Christchurch	Christchurch	20
Christchurch	Dunedin	20
Christchurch	Wellington	15

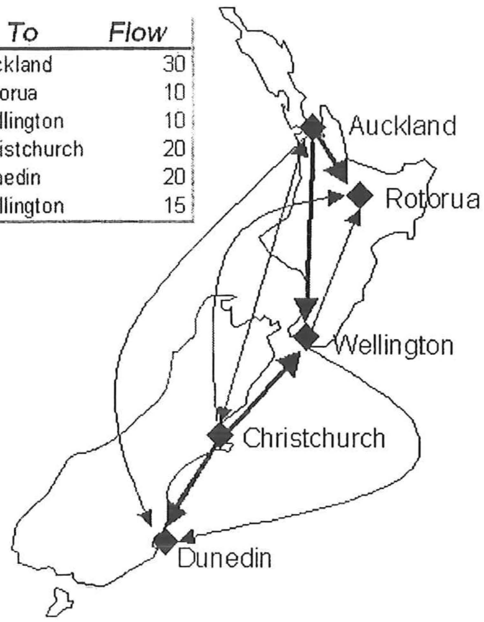

Figure 1.3: Solution to carpet distribution network with linear costs

1.2 Extensions to Pure Network Flow Models

The traditional affine[1] model can be extended in three ways. The first is to add non-linear objective functions to the class of objective functions that can be considered. The second is to allow additional non-network "side constraints". Finally, additional "side variables" can be used to model decisions that cannot be modelled by the traditional network arc structure.

[1] An "affine" function, set, or (sub)space, is simply a linear function, set, or (sub)space translated so that it may or may not include the origin. In contrast, a "linear" function, set, or (sub)space must include the origin by definition.

1.2.1 Nonlinear Objective Functions

A large variety of cost structures and phenomena can be modelled via non-linear functions. One important non-linear cost structure is fixed costs or charges. A "fixed charge" arc can have two costs associated with it. The first cost, called the variable cost, is some function of the flow on the arc. The traditional affine arc cost is an example of this type of cost. The second cost, called the fixed charge, is incurred fully when the flow on the arc is greater than zero, and is not incurred at all otherwise. Fixed charges are often used to model investment and construction costs for plant and equipment. An example of a fixed charge function is presented in Figure 1.4.

Another common real-world cost feature that can be modelled with nonlinear cost functions is economies of scale. Economies of scale arise when the cost per unit flow decreases as the volume or number of units increases. Economies of scale are common in the transportation industry. Figures 1.5 and 1.6 are examples of cost functions displaying economies of scale. The function depicted in Figure 1.5 is a piecewise linear cost function, so called because it is constructed of several connected linear pieces. Such a cost function may arise when a transportation company offers marginal discounts to its customers. For example, a customer may be charged a certain rate per unit for the first 1000 units, a second, lower, rate per unit for the next 500 units, and a third, still lower, rate for any further units over 1500 units. The second function (Figure 1.6) is a more general function that has economies of scale. Such a cost function may be faced by a company in delivering its products to its customers. The cost of transporting the goods per unit of flow may decrease slightly for every extra unit transported (see, for example, Blumenfeld, Burns, Daganzo, Frick & Hall (1987)).

The class of nonlinear cost functions can be logically partitioned into several smaller subclasses. This is useful from an algorithmic perspective as it enables any special structures within a subclass to be exploited. To construct the partitioning, we first make the following definitions.

Definition 1.2.1 (Horst, Pardalos & Thoai (1995)). *Given vectors $\underline{x}_1, \ldots, \underline{x}_m$ in Euclidean space $\mathbb{R}^n$ and real numbers $\lambda_i \geq 0$ with $\sum_{i=1}^{m} \lambda_i = 1$, the vector sum $\sum_{i=1}^{m} \lambda_i x_i$ is called a convex combination of these points.*

Theorem 1.2.1 (Horst et al. (1995)). *A subset C of $\mathbb{R}^n$ is convex iff it contains all the convex combinations of its elements.*

We can then define two important subclasses of the class of general nonlinear functions.

Definition 1.2.2 (Horst et al. (1995)). *A function $\phi : C \to \mathbb{R}$, where C is a convex set in $\mathbb{R}^n$, is called convex if $\phi\left(\lambda \underline{x}_1 + (1-\lambda)\underline{x}_2\right) \leq \lambda \phi\left(\underline{x}_1\right) + (1-\lambda)\phi\left(\underline{x}_2\right)$ for any $\underline{x}_1, \underline{x}_2 \in C$ and $0 \leq \lambda \leq 1$.*

Definition 1.2.3 (Horst et al. (1995)). *A function $\phi : C \to \mathbb{R}$, where C is a convex set in $\mathbb{R}^n$, is called concave if $\phi\left(\lambda \underline{x}_1 + (1-\lambda)\underline{x}_2\right) \geq \lambda \phi\left(\underline{x}_1\right) + (1-\lambda)\phi\left(\underline{x}_2\right)$ for any $\underline{x}_1, \underline{x}_2 \in C$ and $0 \leq \lambda \leq 1$.*

The functions illustrated in Figures 1.4, 1.5, and 1.6 are examples of concave functions. Figure 1.7 depicts an example of a concave function in two variables, whilst Figure 1.8 gives an example of a convex function of two variables. Note that affine functions are both convex and concave.

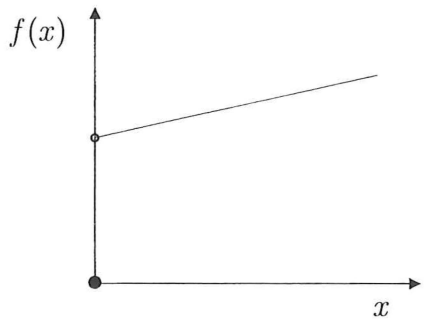

Figure 1.4: A fixed charge function

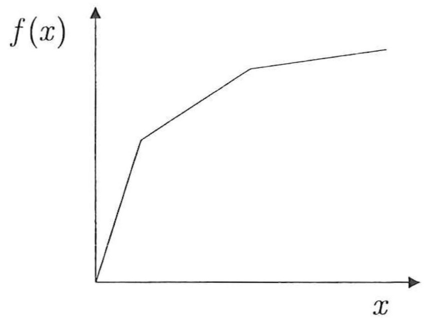

Figure 1.5: A piecewise linear concave function

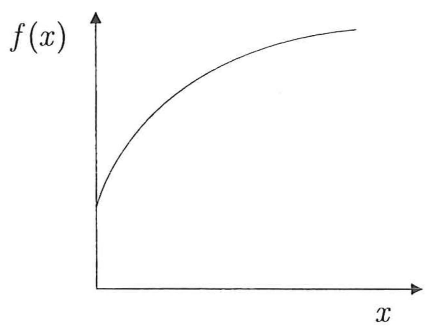

Figure 1.6: A general concave function

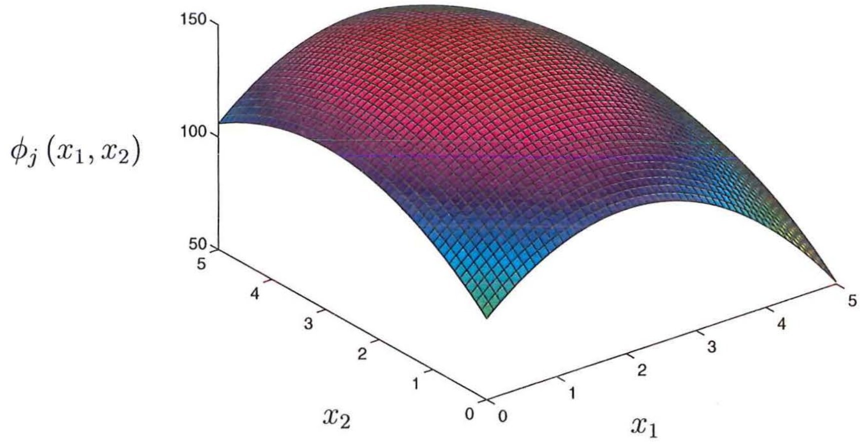

Figure 1.7: Separable concave function of two variables

A function can be non-convex and non-concave. Figure 1.9 graphs such a function.

Nonlinear functions can be further defined as either separable or non-separable. A separable function is defined as follows.

Definition 1.2.4. *A function* $\phi : C \to \mathbb{R}$, *where* C *is a non-empty rectangular set of the form* $C = \{\underline{x} \in \mathbb{R}^n \mid \underline{l} \leq \underline{x} \leq \underline{u}\}$, *is separable on* C *if* $\phi(\underline{x}) = \sum_{j=1}^{n} \phi_j(x_j)$ *where* $\phi_j(x_j)$ *is a function solely of the single variable* x_j *on* $C_j = [l_j, u_j]$.

The functions in Figures 1.8 and 1.9 above are non-separable functions. Conversely, Figure 1.7 is a separable function.

We now return to the carpet distribution network example discussed in Section 1.1. Consider the case where the carpet company is offered a discount by a transport company for all product transported via Wellington. The costs faced by the carpet company to transport directly from the two

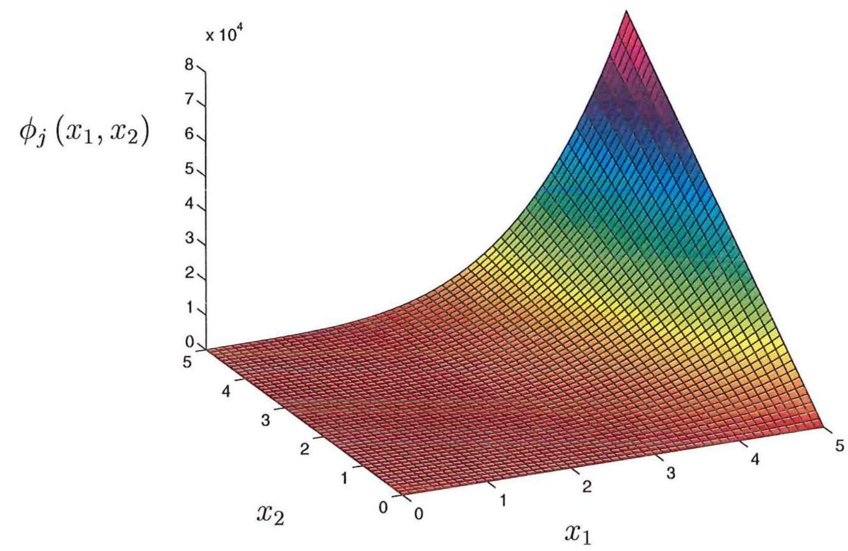

Figure 1.8: Non–separable convex function of two variables

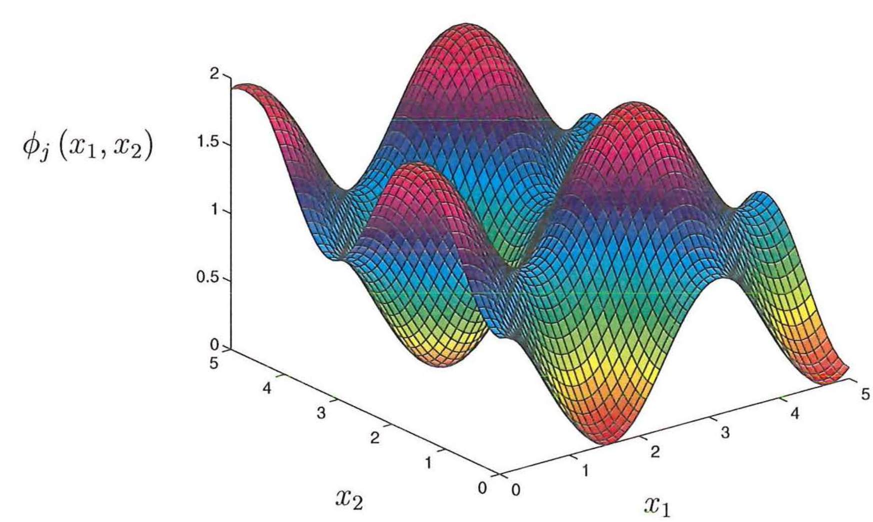

Figure 1.9: Nonseparable nonconcave nonconvex function of two variables

factories to the warehouses excluding Wellington remain the same as before. For carpet shipped to Wellington, the carpet company faces a rate of \$20 per unit for the first 30 units and \$10 per unit thereafter. This cost is calculated on the *total* amount of carpet shipped to Wellington from the two factories. Thus, if 20 units are shipped from each factory, the charge would be \$700 ($= 30 \times 20 + 10 \times 10$). The transport company then charges nothing for goods shipped from Wellington to the other four warehouses.

The diagrammatical network representation of the modified (nonlinear) carpet delivery network is shown in Figure 1.10. Figure 1.11 displays the optimal solution for the problem. Note that it is now worthwhile for the carpet company to ship carpets destined for Rotorua and Dunedin via the Wellington warehouse, rather than directly as before.

1.2.2 Side Constraints

One limitation with traditional network flow models is the inability to model relationships, other than conservation of flow constraints and flow bounds, between arcs. Often however, such "side constraints" exist in and are important to a wide variety of systems. Allowing side constraints to model these relationships therefore permits analysis of a far greater class of problem situations than that able to be modelled by pure network structures.

Regulatory constraints, such as those imposed by government, often impose relationships on otherwise unrelated arcs. For example, the flow on one arc may be regulated to be a within a percentage range of the flow on another arc. Physical constraints on groups of arcs may also exist. For example, a model may contain an arc for the amount of electricity generated

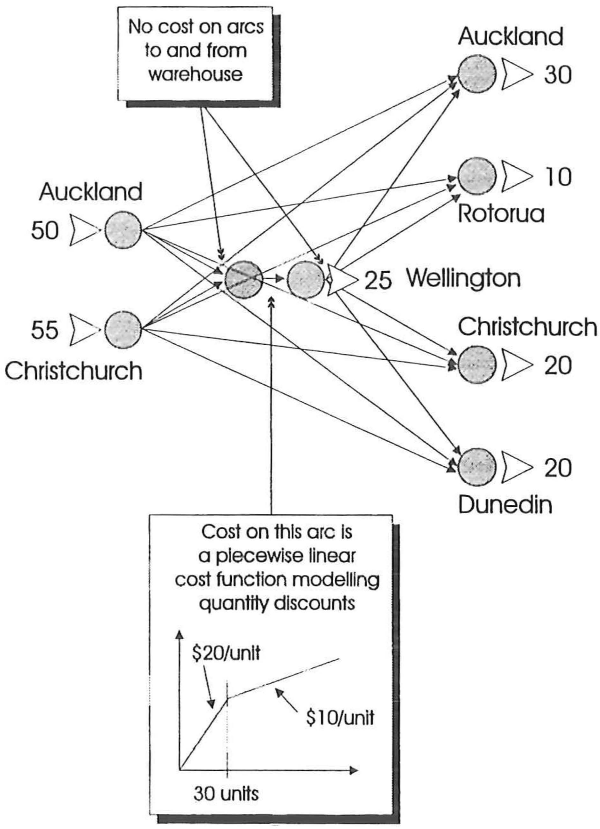

Figure 1.10: Carpet distribution network with nonlinear costs

From	*To*	*Flow*
Auckland	Auckland	30
Auckland	Wellington	20
Christchurch	Christchurch	20
Christchurch	Wellington	35
Wellington	Dunedin	20
Wellington	Rotorua	10

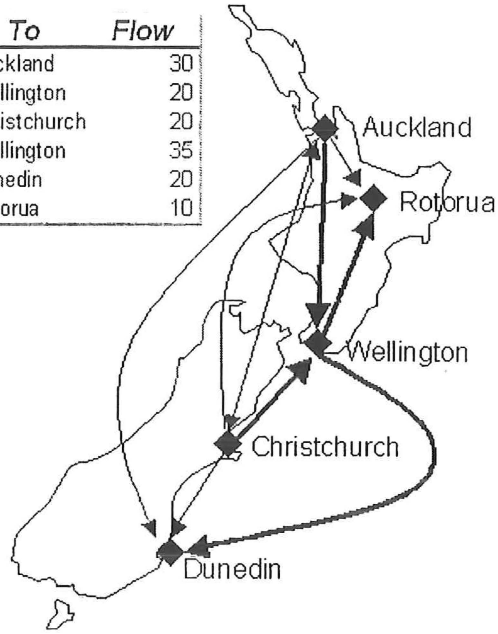

Figure 1.11: Solution to nonlinear carpet distribution network

by a thermal generator in each hour of a twenty four hour time period. Complex physical engineering constraints imposed by the machine, such as restrictions on the rate of change of power generated in successive hourly periods, may then exist. Another example arises in electricity transmission, where Kirchhoff's Laws mean that electrical flow leaving a node does so in certain proportions, dictated by the structure of the transmission network, on each arc. Other examples include gains and losses on arcs, enabling modelling such phenomena as return on investments or electrical power loss on transmission lines, and "bundling" constraints, representing capacities on a set of arcs (rather than on an individual arc).

1.2.3 Side Variables

Often decisions in the system being modelled do not fit naturally into the network structure of arcs and nodes. For example, we may wish to model the generation of electricity by a group of thermal generator units for each hour of a twenty four hour period. The decision to turn a particular unit on or off may be made several times in the model's time horizon. Such a decision, and the associated switching costs, cannot be modelled by the continuous arc flow variables. However, we can use a binary decision variable for each hour period in the twenty four hours that equals zero if the machine is off, and one if the machine is on. The switching costs can then be incurred when the value of this variable changes from one period to the next.

The class of network flow problems considered in this thesis can be extended to allow additional non-network "side" variables. Further, the domain of each side variable may be either continuous or integer. In combination with side constraints, this allows the modelling of a wide variety of feasible regions with mixed continuous and integer decision variables.

1.3 Network Flow Problems Considered in this Thesis

In general, the network flow problems considered in this thesis are characterised by five features: the type of the objective function, the form of the objective function, the type of constraints, the type of the decision variables, and the decision variable domain. These options can be expressed in terms of "drop down menus" as follows:

Objective function type (OFT)	
separable	(S)
non-separable	(N)

Objective function form (OFF)	
linear	(L)
concave	(C)
convex	(X)
arbitrary	(A)

Constraint type (CT)	
node balance	(N)
bounds	(B)
linear side	(S_L)
nonlinear side	(S_N)

Decision variable type (DVT)	
network flow	(F)
side variables	(V)

Decision variable domain (DVD)	
real	(R)
integer	(I)

The class of problem is then specified in the form

$$P\left([OFT], [OFF], [CT], [DVT], [DVD]\right)$$

For example, the linear problem in equation (1.1) is represented as P(S,L,NB,F,R) and the linear and concave problem depicted in Figure 1.10 is of type P(S,LC,NB,F,R). Table 1.2 lists the classes of problems specifically considered in this thesis.

This thesis develops a general method of analysis, called "concave underestimator analysis", for problems of type P(S,LC,NB$S_L S_N$,FV,RI). This analysis develops and uses concave lower underestimators of the separable concave objective functions to construct relaxations of the original problem. A parametric analysis of these relaxations can then be used to provide information on the solution to the original type P(S,LC,NB$S_L S_N$,FV,RI) problem.

An algorithm based on "concave underestimator analysis" is developed in this thesis to solve minimum concave cost network flow problems with linear side constraints and mixed continuous and integer side variables (that is, problems of type P(S,LC,NBS_L,FV,RI). Problems of this type form an important class of problems since a number of applications can be directly modelled as problems of this class. Lamar (1993*b*) showed that in many

Table 1.2: Problems considered in this thesis

Problem Class	Description	Chapters
P(S,LC,NB,F,R)	separable concave cost network flow	3, 4, 5, 7
P(S,LC,NBS_L,F,R)	separable concave cost network flow with linear side constraints	3, 4, 5, 7
P(S,LC,NBS_L,FV,RI)	separable concave cost network flow with linear side constraints and mixed real and integer valued side variables	3, 4, 5, 7
P(S,LC,NB$S_L S_N$,F,R)	separable concave cost network flow with general side constraints	3, 7
P(S,LC,NB$S_L S_N$,FV,RI)	separable concave cost network flow with general side constraints and mixed real and integer valued side variables	3, 7
P(NS,LC,NB$S_L S_N$,FV,RI)	nonseparable concave cost network flow with general side constraints and mixed real and integer valued side variables	7

cases network flow problems involving arbitrary nonconvex arc costs can be converted into equivalent (albeit larger) network flow problems with concave arc costs, thereby enlarging still further the class of applications that can be modelled as problems of type P(S,LC,NBS_L,FV,RI).

Finally, concave underestimator analysis is used to enhance a mixed-integer programming algorithm used to solve the short term electricity dispatch problem. This problem is of the form P(S,L,N,B,S_L,FV,RI).

1.4 Thesis Structure

The remainder of this thesis is organised in the following manner. Chapter 2 reviews the literature on applications of minimum non-convex cost network flow models of type P(S,LC,NB$S_L S_N$,FV,RI). Chapter 3 develops the theory of "concave underestimator analysis" for problems of type P(S,LC,NB$S_L S_N$,FV,RI).

A specific branch–and–bound algorithm for solving mixed integer minimum concave cost network flow problems with linear side constraints and mixed real and integer side variables of type P(S,LC,NBS_L,FV,RI) is presented in Chapter 4. The algorithm applies the concave underestimator analysis presented in the previous Chapter to develop a technique used to accelerate the performance of several algorithms for problems of type P(S,LC,NBS_L,FV,RI), called enhanced capacity improvement. Enhanced capacity improvement is an extension of the capacity improvement technique already present in the literature. Computational testing of the algorithm is reported and analysed in Chapter 5.

Chapter 6 develops a mixed-integer branch and bound algorithm for the short term electricity transmission dispatch problem that uses the capacity improvement techniques of Chapter 4.

Finally, Chapter 7 summarises the content of the thesis and contains a discussion of possible extensions to the concave underestimator analysis developed in Chapter 3 and the capacity improvement techniques developed in Chapter 4.

Chapter 2

Solution Methods and Applications

2.1 Introduction

Determining the global optimal solution to problems of type P[SNS,LCXA,NB$S_L S_N$,FV,RI] considered in this thesis is challenging, because a local optimal point is not necessarily a global one; and the number of local optimal points can be enormous, even for moderate-sized problems. Even the relatively "simple" subclass of minimum concave cost network flow problems involving strictly concave objective functions is known to be NP-hard (Guisewite & Pardalos 1990).

There is a rich literature that identifies potential applications and proposes solution approaches for problems of type P[SNS,LCXA,NB$S_L S_N$,FV,RI]. Excellent articles and texts reviewing both solution algorithms and methodologies, and to a lesser extent potential application areas, abound (see, for example, the surveys in Rinnooy Kan & Timmer (1989), Guisewite & Pardalos (1990), Horst (1990), Benson (1995), and Benson (1996), and the treatments given in Horst et al. (1995) and Horst & Tuy (1996)). Unfortunately due to the complexity of problems of type P[SNS,LCXA,NB$S_L S_N$,FV,RI], and the consequent lack of fast, accessible algorithms for solving such problems, actual reported application of models of type P[SNS,LCXA,NB$S_L S_N$,FV,RI] to real world problems has been limited.

Rather than repeat this comprehensive survey literature, in the following Chapter we first present a broad overview of the literature on solution methods for problems of type P[SNS,LCXA,NB$S_L S_N$,FV,RI]. We then provide a more detailed discussion on both potential and actual applications of problems of type P[SNS,LCXA,NB$S_L S_N$,FV,RI], with emphasis on the latter, that have been reported in the literature.

2.2 Solution Methods

The solution procedures presented in the literature can be broadly classified into four groups: branch and bound algorithms; enumerative procedures; outer approximation; and convexification. These four groups are not mutually exclusive – many algorithms may be classified into more than one group. Often these procedures are designed to solve a subclass of problems of type P[SNS,LCXA,NB$S_L S_N$,FV,RI] only.

A solution procedure widely used in the literature to solve continuous global optimisation problems is branch and bound. Similar to the integer programming branch and bound technique of Land & Doig (1960), branch and bound procedures partition the feasible region into successively smaller subregions. Lower bounds to the optimal objective function value of the problem over these subregions, and upper bounds to the optimal objective function value of the original problem, are then determined. Regions that have a lower bound greater than the incumbent (i.e. best known) upper bound to the optimal solution value of the original problem are removed from consideration, as they cannot contain a solution better than the incumbent. The remaining regions are then partitioned still further, and the process repeats until either the difference between the minimum of the lower bounds and the incumbent upper bound is smaller than some predetermined tolerance or there are no unfathomed subregions remaining.

Branch and bound algorithms for the minimisation of separable concave functions over a compact convex feasible region were first developed by Falk & Soland (1969). Following this seminal paper, Horst (1976) considered problems involving more general non-separable continuous non-convex cost functions. Horst's paper extended the rectangular partitioning branch and bound algorithm of Falk & Soland (1969) by considering more general simplicial partitions, partitioning strategies, and lower bounding strategies. Subsequently, a large number of branch and bound algorithms for continuous global optimisation have appeared in the literature (see, for example, Soland (1974), Benson (1982), Tuy & Horst (1988), Benson (1990), Thakur (1990), Benson & Horst (1991), Benson & Sayin (1994), Shectman & Sahinidis (1998), and Ryoo & Sahinidis (1996), and the treatments given in Pardalos & Rosen (1987), Horst (1990), Rinnooy Kan & Timmer (1989), Horst et al. (1995), Benson (1995), Benson (1996), and Horst & Tuy (1996)).

Branch and bound algorithms have also been proposed for concave pure integer or mixed integer optimisation. Benson, Erenguc & Horst (1990) show how solution procedures for continuous global optimisation can be extended to the discrete case by considering the class of branch and bound algorithms. They develop a prototype general branch and bound algorithm to solve integer global minimisation problems which is a modification of the general prototype algorithms presented in the literature for solving continuous concave minimisation problems. They then develop an implementation of their prototype algorithm for nonseparable concave integer minimisation over a polytope. Benson & Erenguc (1990) and Bretthauer, Cabot & Venkataramanan (1994) develop similar branch and bound algorithms for integer global minimisation problems. Each of the three approaches differ in the details of the construction of convex envelopes, subproblem construction, and the partitioning schemes used. Cabot & Erenguc (1986) also develop a branch and bound algorithm for the problem of separable concave integer minimisation over a polyhedron. Finally, Al-Khayyal & Larsen (1990) proposed a branch and bound algorithm involving sequences of linear programming subproblems to solve mixed integer quadratic minimisation problems.

Enumerative techniques are predicated on the fact that the optimal solution to problems with concave cost functions over a compact convex feasible set occurs at an extreme point or vertex of the feasible region (e.g. Horst et al. (1995)). However, complete enumeration of the entire set of feasible points is computationally prohibitive for even small problems. Hence various procedures that attempt some form of implicit enumeration or intelligent search of the extreme point set have been proposed in the literature.

Florian & Robillard (1971) propose an algorithm for the concave cost network flow problem that is based on the equivalence of the original network flow problem to an uncapacitated bipartite network. A branch and bound structure is then used to conduct an implicit enumeration of the set of extremal flows. Extreme point ranking procedures, first proposed by Murty (1969) for fixed charge problems, rank the extreme points of the feasible region. Starting with the vertex that minimises some (linear) objective function relaxation of the original problem, a search is conducted of adjacent vertices of the current ranked vertex to find the "next-ranked" vertex. The search is terminated when known lower bounds to the optimal solution at those vertices adjacent to the current vertex are larger than the current optimal solution upper bound. Enumerative extreme point ranking algorithms are presented in Taha (1973) and McKeown (1975).

Cutting plane algorithms, another enumerative technique, have evolved from the work of Tuy (1964). In this paper, Tuy developed a "cut", or additional constraint, that can be used to eliminate part of the feasible region of the problem. Glover (1973) then extended this idea to more general "convexity cuts". Cabot (1974) and Horst & Tuy (1996) propose algorithms that generate sequences of subproblems via the application of cuts. Such algorithms effectively search through the extreme points of the feasible region, removing "candidate" vertices via the use of the cuts. Algorithms may also use cutting planes as part of a more general over–all solution strategy. Bretthauer & Cabot (1994) present an algorithm for concave minimisation over a polyhedron that employs Tuy cuts in a branch and bound search strategy.

In an interesting application of cutting plane theory, Bretthauer (1994) develops penalties based on the Tuy cutting plane for use in a branch and bound solution strategy. The penalties assist in fathoming nodes in the branch and bound tree, thereby reducing the number of subproblems required during the branch and bound search. The Tuy cutting plane is never explicitly added to the constraint set, thereby preserving any structure, such as a pure network, inherent in the feasible region of the problem.

A concept similar to cutting planes is that of valid inequalities. An inequality is called *valid* for a set if it holds for all elements of that set. Valid inequalities can be used in integer programming to remove part of the feasible region of a relaxation of the integer program, without removing any solutions that are feasible to the original integer program. The Gomory cut ((Gomory 1963)) is an early example of a valid inequality. Algorithms incorporating valid inequalities are common in the literature, and are treated fully in Nemhauser & Wolsey (1988) and Nemhauser & Wolsey (1989).

Cone covering algorithms were also proposed in Tuy (1964). These algorithms construct a set of cones that cover or partition the feasible region still to be explored. By creating and solving a linear program for a particular cone, the algorithm searches the extreme points of that part of the feasible region covered by that cone. If it can be determined that the feasible region covered by the cone does not contain a better solution than the current incumbent, the cone is discarded. If a new incumbent solution is found, part of the feasible region containing the previous incumbent solution is removed via a cut, a new cone is constructed that covers the reduced feasible region, and the algorithm restarts. Alternatively, if it is unknown whether or not a cone contains a better solution than the incumbent, it is partitioned into two or more cones. Zwart (1974), Jacobsen (1981), and Horst & Tuy (1996) propose cone covering algorithms of this type.

Outer approximation methods also involve the use of cuts. A relaxation of the original feasible region is constructed, over which the original cost function is minimised. If the solution is feasible to the original problem, the solution procedure is terminated. Otherwise, a cut is constructed that removes part of the relaxed feasible region that contains the current candidate solution. The cut is specially constructed so as not to exclude any part of the original feasible region. The process is then repeated until a solution to the original problem (to some predetermined level of accuracy) is found. Horst & Tuy (1996) provide a detailed development and analysis of outer approximation methods. Benson (1990) develops an interesting partial branch and bound–outer approximation hybrid algorithm for minimising separable concave functions over a compact convex set. The advantages of this approach over standard outer approximation are two–fold: it does not expand the problem size via the addition of explicit cuts as much as outer approximation; and it does not require explicit construction of polyhedra that contain the feasible region. The algorithm of Falk & Hoffman (1976) is essentially a variant of the outer approximation approach for non–separable concave minimisation problems. It generates a sequence of linear programs that are formed by effectively generating the convex envelope of the original objective function over a set that contains the feasible region. This containing set is then successively tightened to obtain a more refined envelope. Hoffman (1981) extends this approach to general convex constraint sets.

Finally, an alternative (and relatively uncommon) approach for the solution of non-convex minimisation problems is that of convexification. There is a rich literature for primal-dual lagrangian solution methods (such as steepest ascent and Newton algorithms) for the solution of problems that are locally or globally convex in structure. Unfortunately, such methods fail when their convexity conditions are violated. However, problems with a non-convex objective can be "convexified" via a the addition of a penalty function to create a function that has at least a locally convex structure. The resulting problem can then be solved using the standard convex optimisation techniques. Several methods for achieving the convexification have been presented in the literature. Examples include penalties based upon the euclidean norm of some vector valued function, such as the constraint equations or some other function of the decision variables. Bertsekas (1979), and Horst (1984) provide detailed expositions of this methodology.

2.3 Applications

2.3.1 Waste-Disposal and Management Systems

A common area of application of concave cost network models has been waste–disposal and management systems. Waste management systems are usually structured into four phases: collection, transportation, processing, and disposal. The collection of the waste depends upon its nature; for example urban solid waste is often collected door-to-door by trucks, and urban liquid waste is collected by the household sewerage system. The waste is then transported to disposal sites either directly or through intermediate transfer stations where it can be processed in a variety of ways.

The structure of waste-disposal and management systems lends itself to modelling as a network or generalised network with side-constraints. The costs involved in the expansion and operation of the network are also often non-linear in nature. This non-linearity can consist of a large fixed cost component for the purchase of land, construction of plant, and purchase of equipment, and a constant or decreasing marginal cost component for plant operation and waste transportation due to economies of scale. Therefore, models for waste systems of type P[SNS,LCXA,NBS_LS_N,FV,RI] are relatively common in the literature and can be used to investigate two important aspects of waste–disposal and management systems. The first aspect deals with the design of the network: how many, what type, and where should transfer stations and disposal sites be located? The second deals with the scheduling or coordination of product flows within the network.

Bloemhof-Ruwaard, Salomon & van Wassenhove (1996) study and present a general formulation for problems of this type with fixed charge cost functions. They develop a branch–and–bound solution algorithm using "valid inequalities", and test this algorithm on a series of random test problems.

Jarvis, Rardin, Unger, Moore & Schimpeler (1978) model the optimal design of regional waste-water systems. Regional waste-water systems consist of collector nodes, where waste water from a local population is collected before being transported along a system of pipes to a treatment plant site.

Due to economies of scale, the cost of building new pipes and treatment plants is concave. The expansion of existing or the design of new waste-water systems, containing both existing and proposed treatment plants and pipes, was modelled in Jarvis et al. (1978) as a network flow problem with piece-wise linear concave costs. They used this approach to develop a model of waste-water systems for the North County Basin of Jefferson County, Kentucky, U.S.A. Jarvis et al. (1978) report that the model was used heavily for "what-if" type analyses for the design of waste-water systems in the North County Basin area.

The problem of citing intermediate transfer stations in a solid-waste transport system is considered in Khan (1987). They model this problem as a fixed cost generalised transportation network. As in Jarvis et al. (1978), the model can be used for "what-if" analyses given current or proposed waste transport systems, to cost proposed transfer stations, and to determine minimise transportation costs for given physical network structures. They then demonstrate their model on a composite example based on the data from several waste management programmes.

The solid-waste management problem is also examined by Caruso, Colorni & Paruccini (1993). The system studied is very similar to that examined in Khan (1987). However, the citing of disposal sites as well as intermediate transfer stations are considered. In addition, Caruso et al. (1993) consider not just the economic costs of facility construction and operation, but also the costs associated with the waste of resources and environmental impact. They model this as a fixed charge multi-objective problem. They then applied their model to the solid-waste management system of the Lombardy region in France.

2.3.2 Cotton-Ginning

Klingman, Randolph & Fuller (1976) examine the plant location problem of the cotton-ginning industry, and in particular that of the Mesilla Valley of New Mexico and the Upper Rio Grande Valley of El Paso County, Texas. In the mid-70's, the U.S. cotton industry began experiencing an excess gin plant capacity. In addition, innovations in cotton storage methods permitted the cotton farmers to store harvested cotton rather than send it to a cotton gin immediately for processing. Therefore, the problem was one of designating which gins should be activated for processing cotton, and how much cotton each farm should ship to which gin in each week of the season.

Opening a gin incurred a large fixed cost, which included the cost of electrical connection. The electrical connection charge also purchased a quantity of electric power that processed a volume of cotton, after which power was purchased at the normal price. Therefore the cost associated with opening and operating a gin was convex piecewise linear with a fixed charge, and the problem was modelled in Klingman et al. (1976) as a transportation network with fixed charges.

2.3.3 Determining Optimum Cast Bloom Lengths

A bloom is a rectangular piece of steel that is rolled on a finishing mill into a structural shape. Large "as-cast" blooms are rolled and then cut into "as-rolled" blooms of smaller cross Section, before being placed in a reheat furnace for rolling on a finishing mill into the desired structural shapes. Waste occurs when an "as-cast" bloom length does not contain a perfect integer multiple of an "as-rolled" bloom length. The problem to overcome in this process is to attempt to minimise the number of different "as-cast" bloom lengths of different grades required whilst trying to minimise the waste from the "as-cast" blooms.

Vasko, Newhart, Stott & Wolf (1996) construct an uncapacitated fixed-charge facility location problem. In their model the cost of casting and storing each potential "as-cast" bloom length is modelled as a fixed cost. Waste is penalised via a linear variable cost. The model can then be used to select a sub-set of the candidate "as-cast" blooms. Vasko et al. (1996) then report an application of their model that reduced the number of "as-cast" bloom lengths required for a particular manufacturing situation from 660 to 65.

2.3.4 Allocation of Launch Vehicles to Space Missions

The problem examined in Stroup (1967) is to aid selection of launch vehicle designs for development and to assign planned and existing vehicles to minimise the cost of performing future space missions. Stroup (1967) models this problem as a fixed charge assignment problem, where the fixed charge represents the expected development costs of a vehicle and the costs of performing a mission with a particular vehicle is modelled as a linear variable cost. Stroup (1967) reports that the model has been used on problems involving from 30 to 70 candidate and 10 existing vehicles and about 300 missions.

2.3.5 Distribution and Transportation

The problem of determining the routes and facilities, and consequent transportation schedules, to use for product distribution and transportation often exhibit cost structures with economies of scale. This, along with the inherent network structure of such problems, means that such problems fall naturally into the class of problems of type P[SNS,LCXA,NB$S_L S_N$,FV,RI].

Klincewicz (1990) examines the situation where products from various sources can be shipped to the either directly to their destinations, or via "consolidation terminals". They model this problem as a concave cost multi–product network flow and develop optimal and heuristic solution techniques that involve decomposing the problem into a series of concave cost facility location problems. They then test their methods on test problems based on the locations of major cities in the US.

The logistics systems at General Motors (GM) is examined in Blumenfeld et al. (1987). The product distribution system of a "typical" division at GM consisted of a few manufacturing plants that shipped product via several warehouses to GM assembly plants located throughout the USA. The major costs involved in these distribution networks were the cost of transporting the goods through the system, and the costs associated with keeping inventories at the manufacturing and warehouse locations. Due to economies of scale, the transportation costs were concave. The problem became one of minimising the combined corporate costs of inventory and transportation for the divisions products. Blumenfeld et al. (1987) model of the Delco Electronics Division's product distribution system identified logistics savings in the order of 26% of the division's original logistics cost (approximately $2.6 million). They also discuss implementation of similar models throughout General Motors.

2.3.6 Designing Offshore Natural Gas Pipeline Systems

Impurities in offshore natural gas are separated from the gas at a land based separation plant. The separation plant typically serves a number of wells. The problem is to design a cost–efficient pipeline system to transport the gas from the offshore wells to the separation plant. As was the case in the problem of designing waste-water distribution systems presented in Jarvis et al. (1978), the cost of the pipes has a fixed component independent of the capacity and a component that is a function of capacity that can be either linear or concave due to economies of scale. Such problems are considered in Hochbaum & Segev (1989) and Rothfarb, Frank, Rosenbaum & Steiglitz (1970).

2.3.7 Communication Network Design

Communications networks can be modelled easily as network flow problems, with a set of nodes (traffic points) and two-way links (transmission systems) connecting the nodes. The cost of the network arises largely from providing links and the number of channels on each link. Transmission systems that can be installed on a link display economies of scale. That is, the marginal cost of an extra channel decreases as the number of channels on the link increases. Therefore, the cost functions involved are concave, and the system can be modelled via models of type P[SNS,LCXA,NB$S_L S_N$,FV,RI].

Yaged (1971) analyses the minimum cost routing problem for concave networks. Given a communications network and a flow profile detailing the flow between from node i to node j in the network, the problem is to determine the optimal routing for the network, and therefore the channels required on each link. A local optimisation procedure is developed in Yaged (1971), and applied to a simplified model of the US long-distance telecommunications network. This problem is also discussed in Hochbaum & Segev (1989).

The routing problem can also be extended to network design; that is, determining which links to build. Potential links can be included in the routing problem, with associated fixed construction and set-up costs. Proposed links that are used in routing the flow in the network can then be considered for construction. Such models can be useful for "what-if" type analyses.

Fetterolf & Anandalingam (1992) examine the problem of interconnecting a group of local area networks (LANS) with bridges. The objective of the design is to minimise the cost of interconnection whilst maintaining acceptable traffic intensity levels on the LANS and bridges. They formulate this problem as a special case of a fixed charge multi-commodity network with side constraints. A Lagrangian relaxation algorithm for solving the LAN interconnection problem is also presented.

2.3.8 Analysis of the Impact of Industry Regulation

In an interesting application, Kumar (1988) discusses the application of models of type P[SNS,LCXA,NB$S_L S_N$,FV,RI] to the impact analysis of governmental regulatory decisions. They develop their methodology in terms of the chemical industry. A fixed charge model of the industry, including investment decisions, is constructed. Governmental regulation, such as a restriction on the production of a particular output, is then incorporated in the model's constraints. The model can then be solved and analysed, and compared to the model of the non-regulated situation. The effect and impact of the regulation can then be evaluated.

Kumar (1988) develop an illustrative fixed-charge model of the chemical processing and distribution industry. They then use this model to evaluate and analyse the effects of five forms of regulation under the Toxic Substances Control Act (US).

Chapter 3

Concave Underestimator Analysis

3.1 Introduction and Motivation

Many of the solution approaches discussed in the previous Chapter for problems of type P(S,LC,BNS_LS_N,FV,RI) involve the use and solution of "relaxations" of (some part of) the original problem. These relaxations generally involve either a simpler objective function or a simpler (and larger) feasible region than in the original problem. For example, outer approximation algorithms involve the construction of larger feasible regions that contain the part of the feasible region of the original problem being searched. The relaxed feasible region is constructed so that minimisation of the objective function over this new feasible set is comparatively simple. The branch and bound solution methodology solves a relaxation of the original problem at each branch and bound node in which the objective function of the original

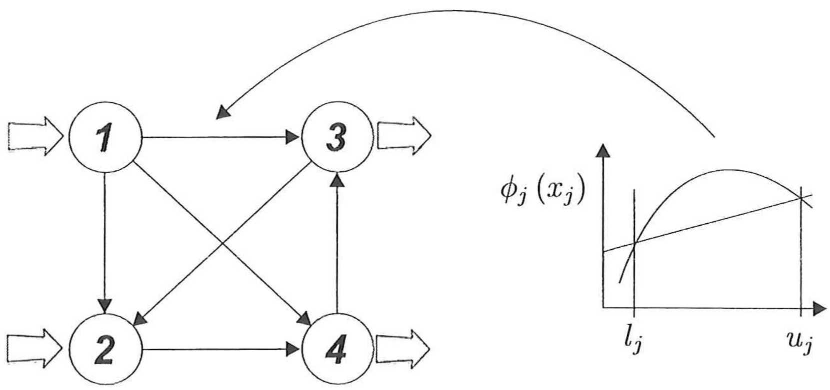

Figure 3.1: "Typical" network flow problem P with concave arc costs

problem is replaced with its convex envelope. The solution to this relaxation provides a lower bound to the solution of the subproblem at that node of the branch and bound tree.

To illustrate this idea, consider the "typical" network presented in Figure 3.1, which we denote as problem P. Each arc in the network has associated lower and upper flow bounds and an associated (separable) concave objective function. The branch and bound approach for solving this problem is illustrated general terms in Figure 3.2. A relaxation of problem P can be formed by replacing the concave objective function on each arc with its (affine) convex envelope defined between the lower and upper flow bounds for that arc. The relaxed problem is a linear minimum cost network flow problem, and is easily solved. The optimal solution to the relaxed problem provides a lower bound to the optimal solution of the original problem P. Next, for some given arc j two new subproblems are formed (denoted Q_1 and Q_2) by partitioning or bisecting the feasible region into two subregions at some flow value, denoted $\tilde{x}_j$, on arc j. Q_1 is formed from the feasible

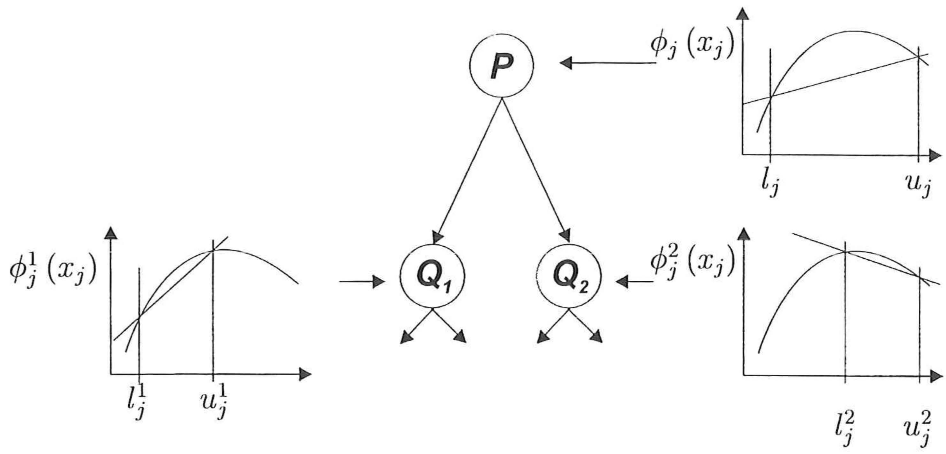

Figure 3.2: "Typical" branch and bound solution tree

region of the original problem Q with the addition of the constraint $x_j \leq \tilde{x}_j$, and Q_2 is formed from the feasible region of the original problem Q with the addition of the constraint $x_j \geq \tilde{x}_j$. Note that $\tilde{x}_j$ is defined such that the feasible regions for Q_1 and Q_2 are not empty. In problem Q_1, the convex envelope of the objective function for arc j is tighter (that is, closer to the original concave objective) over the range of feasible flows for arc j than the original convex envelope. Similarly, in problem Q_2, the convex envelope of the objective function for arc j is tighter (that is, closer to the original concave objective) over the range of feasible flows for arc j than the original convex envelope. Thus, the minimum of the solutions to the relaxed problem at Q_1 and Q_2 provides a tighter (that is, better) lower bound to the optimal solution to P than the previous lower bound. This general process continues until a solution is found of the desired accuracy.

One disadvantage of the "objective relaxation" solution approach using convex envelopes is that the shape or behaviour of the cost function is not

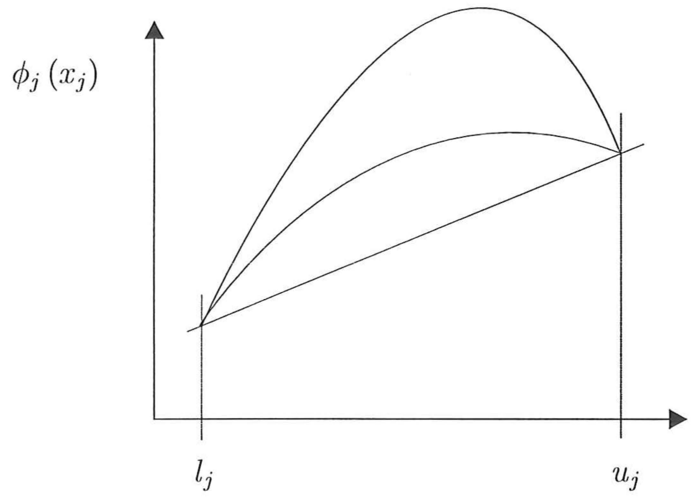

Figure 3.3: Two different concave functions with same convex envelope

explicitly taken into account. Specifically, for a concave function of one variable, the convex envelope over a particular range is the affine function that coincides with the original concave function at the end points of that range. The behaviour of the objective function between the lower and upper flow bounds is ignored. For example, both concave objective functions depicted in Figure 3.3 have exactly the same convex envelope.

The contribution of this Chapter is the development of "concave underestimators". Similar in concept to convex envelopes, concave underestimators take account of the shape of the concave objective function between the lower and upper arc flow bounds. First, we introduce some notation used throughout the Chapter, and define the concept of "relaxations". We then present the traditional relaxation of problems of type P(S,LC,BN$S_L S_N$,FV,RI) based on convex envelopes. Next, the concept of concave underestimators is defined, and a relaxation of problems of type P(S,LC,BN$S_L S_N$,FV,RI) based on concave underestimators is developed.

Finally, the concave underestimator relaxation is used to develop a powerful method of parametric analysis for problems of type P(S,LC,BN$S_L S_N$,FV,RI).

3.2 Notation

In this chapter we are concerned with problems of type P(S,LC,BN$S_L S_N$,FV,RI). For convenience, we denote the problem of type P(S,LC,BN$S_L S_N$,FV,RI) currently under consideration as problem Q. That is, problem Q is of the form

$$
\begin{array}{llll}
(\mathrm{Q}) & \min & \phi\left(\underline{x}\right) & \\
 & \text{s.t.} & A\underline{x}_{\mathbb{R}} = \underline{b} & \\
 & & \underline{\gamma}\left(\underline{x}\right) = \underline{g} & \\
 & & l_j \leq x_j \leq u_j & \forall j \in J \\
 & & \underline{x} = \left(\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}\right) \in \mathbb{Y}^{n+m} &
\end{array}
\tag{3.1}
$$

where $\underline{x} = \left(\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}\right) = \left(\ldots, x_j, \ldots\right)^{\mathrm{T}}$ is a vector of decision variables with index set $J = \{1, \ldots, j, \ldots, n+m\}$, $\mathbb{Y}^{n+m}$ is the subspace of $\mathbb{R}^{n+m}$ such that $\mathbb{Y}^{n+m} = \{\mathbb{R}^n; \mathbb{Z}^m\}$, $\underline{x}_{\mathbb{R}} \in \mathbb{R}^n$ is the vector of continuous valued solution variables, $\underline{x}_{\mathbb{Z}} \in \mathbb{Z}^m$ is the vector of integer valued solution variables, $\underline{l} = \left(\ldots, l_j, \ldots\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ is the lower bound vector for the decision variable vector $\underline{x}$, $\underline{u} = \left(\ldots, u_j, \ldots\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ is the upper bound vector for the decision variable vector $\underline{x}$, A is the matrix of coefficients for the network conservation of flow constraints (the node-arc incidence matrix), $\underline{\gamma}\left(\underline{x}\right)$ is the vector of real valued (possibly nonlinear) functions on $\underline{x}$ and $\underline{g}$ the vector of real valued scalars that together define the side constraints of problem Q, and $\phi(\underline{x})$ is a separable concave real-valued function defined on the decision variables $\underline{x}$ that performs the mapping $\phi : \mathbb{R}^{n+m} \to \mathbb{R}$.

For convenience, a compact notation for describing this general class of problems can be formed as follows. We define S as the set of feasible solution vector values defined by the constraints of problem Q

$$S = \left\{ \underline{x} = (\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}) : \left\{ \begin{array}{ll} A\underline{x}_{\mathbb{R}} & = \underline{b} \\ \underline{\gamma}(\underline{x}) & = \underline{g} \\ \underline{l} \leq \underline{x} \leq \underline{u} & \end{array} \right\} \right\}$$

The set S can be further partitioned into $S = X \cap H$, where

$$X = \left\{ \underline{x} = (\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}) : \left\{ \begin{array}{l} A\underline{x}_{\mathbb{R}} = \underline{b} \\ \underline{\gamma}(\underline{x}) = \underline{g} \end{array} \right\} \right\}$$

and H is the hyperrectangle defined by the simple flow bounds on $\underline{x}$ as follows

$$H = \{ \underline{x} : \underline{l} \leq \underline{x} \leq \underline{u} \}$$

We can then write problem Q as

$$(Q) \qquad \min \phi(\underline{x}) \text{ s.t. } (\underline{x}) \in S = X \cap H, \underline{x} = (\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}) \in \mathbb{Y}^{n+m}$$

In addition, throughout this thesis for any problem $\bullet$, let $\nu[\bullet]$ denote the optimal objective function value of $\bullet$. Also, for any problem $\bullet$, let $lb[\bullet]$ denote a lower bound to $\nu[\bullet]$, and let $ub[\bullet]$ denote an upper bound to $\nu[\bullet]$.

3.3 Relaxations

The analysis of problem Q is based upon the construction of both linear and nonconvex relaxations of Q. Formally, a relaxation of Q can be any problem that satisfies the following two properties (Geoffrion & Marsten 1972):

(R1) The feasible region of the relaxation contains or equals $S \cap \mathbb{Y}^{n+m}$;

(R2) The objective function of the relaxation at each point $\underline{x} \in S \cap \mathbb{Y}^{n+m}$ is less than or equal to $\phi(\underline{x})$.

For our purposes, it is convenient to distinguish between these two properties. Therefore, we define a *feasibility relaxation* of Q as any relaxation whose feasible region strictly contains $S \cap \mathbb{Y}^{n+m}$. We similarly define an *objective function relaxation* of Q as any relaxation whose objective function does not equal $\phi(\underline{x})$ at each point $\underline{x} \in S \cap \mathbb{Y}^{n+m}$.

3.4 Convex Envelopes

We begin with the following formal definition of a convex envelope:

Definition 3.4.1 (see Horst et al. (1995)). *Let $f : S \to \mathbb{R}$ be a lower semi-continuous function, where S is a nonempty convex set in $\mathbb{R}^n$. The* convex envelope *of $f(\underline{x})$ taken over S is a function $\bar{f}(\underline{x})$ such that*

(i) $\bar{f}(\underline{x})$ *is convex on S.*

(ii) $\bar{f}(\underline{x}) \leq f(\underline{x}) \ \forall\, \underline{x} \in S$.

(iii) If $h(\underline{x})$ is any convex function defined on S such that $h(\underline{x}) \leq f(\underline{x})\, \forall \underline{x} \in S$, then $h(\underline{x}) \leq \bar{f}(\underline{x})\, \forall \underline{x} \in S$.

We can now specify a relaxation of Q that uses convex envelopes. We first define the set $\bar{X} \subset \mathbb{R}^{n+m}$ as follows:

$$\bar{X} = \left\{ (\underline{x}) : \left\{ \begin{array}{c} A\underline{x} = \underline{b} \\ C\underline{x} = \underline{g} \end{array} \right\} \right\} \tag{3.2}$$

where C is a matrix of elements $c_{ij} \in \mathbb{R}$, $\underline{g}$ is a vector of real valued scalars, and $C\underline{x} = \underline{g}$ is defined such that $X \subseteq \bar{X}$. We then define $\bar{S} \in \mathbb{R}^{n+m}$ such that $\bar{S} = \bar{X} \cap H$. Note that $S \subseteq \bar{S}$.

The convex envelope relaxation of Q presented here, denoted $\bar{Q}$, is both an *objective* and *feasibility* relaxation of problem Q:

$$(\bar{Q}) \qquad \min \bar{\phi}(\underline{x}) \text{ s.t. } \underline{x} \in \bar{S} = \bar{X} \cap H,\ \underline{x} \in \mathbb{R}^{n+m}$$

where $\bar{\phi}(\underline{x})$ is the convex envelope of $\phi(\underline{x})$ on H. Because of the separability of $\phi(\underline{x})$, we have $\bar{\phi}(\underline{x}) = \sum_{j \in J} \bar{\phi}_j(x_j)$ where $\bar{\phi}_j(x_j)$ is the convex envelope of $\phi_j(x_j)$ over the range $l_j \leq x_j \leq u_j$. Note that because the objective functions considered here are either separable linear or separable concave functions, their convex envelopes are in fact affine (linear) functions. That is

$$\bar{\phi}_j(x_j) = f_j + c_j \cdot x_j \tag{3.3}$$

where

$$\begin{aligned} c_j &= \frac{\phi_j(u_j) - \phi_j(l_j)}{u_j - l_j} \\ f_j &= \phi_j(l_j) - c_j \cdot l_j \end{aligned} \tag{3.4}$$

Since each $\bar{\phi}_j(x_j)$ is affine and $\bar{S}$ is a polytope in $\mathbb{R}^{n+m}$, problem $\bar{Q}$ is a linear program and is easily solved. Hence, problem $\bar{Q}$ is referred to as the *linear relaxation* of problem Q.

Let $\underline{\bar{x}} = (\ldots, \bar{x}_j, \ldots) \in \mathbb{R}^{n+m}$ be the optimal solution vector for problem $\bar{Q}$, let a_{ij} denote the element in the i-th row and j-th column in the simplex tableau of problem $\bar{Q}$, let $\bar{a}_{ij}$ denote the element in the i-th row and j-th column in the *optimal* simplex tableau of problem $\bar{Q}$, and let $\bar{\pi}_i$ denote the dual variable associated with the i-th row of the optimal simplex tableau of problem $\bar{Q}$. The reduced cost associated with variable x_j in the optimal solution to problem $\bar{Q}$, denoted $\bar{r}_j$, is given by $\bar{r}_j = c_j - \bar{z}_j$, where $\bar{z}_j = \sum_{i \in I} \bar{\pi}_i \cdot a_{ij}$. In addition, we partition the index set J into $J = B \cup NL \cup NU$ where $j \in B$ if x_j is a basic variable, $j \in NL$ if x_j is a nonbasic variable at its lower bound l_j, and $j \in NU$ if x_j is a nonbasic variable at its upper bound u_j in the optimal solution to problem $\bar{Q}$. The optimal solution to problem $\bar{Q}$ is used in defining the nonconvex relaxation of Q discussed in the following Section.

3.5 Defining the Concave Underestimator

In this Section, we define the concept of concave underestimators, and develop a nonlinear relaxation of problem Q that utilises concave underestimators. The definition of the concave underestimator of a separable concave function is problem specific. In general, the specification of the concave underestimator $\hat{\phi}_j(x_j)$ of the concave objective function $\phi_j(x_j)$ associated with variable j is predicated on the optimal solution to the linear relaxation $\bar{Q}$. To define $\hat{\phi}_j(x_j)$, it is useful to define another function, referred to as the *reduced cost function*, associated with $\hat{\phi}_j(x_j)$. The reduced cost function for any $j \in J$ is denoted $\Delta_j(x_j)$ and is defined as

$$\Delta_j(x_j) = \hat{\phi}_j(x_j) - \bar{z}_j \cdot x_j \tag{3.5}$$

The reduced cost *function* is a straight forward extension of the reduced cost *coefficient* used in linear programming. In fact, if $\hat{\phi}_j(x_j)$ is affine, then $\Delta_j(x_j) = \bar{r}_j \cdot x_j$ where $\bar{r}_j$ is the reduced cost coefficient associated with variable j in the optimal solution to problem $\bar{Q}$.

The concave underestimator of $\phi_j(x_j)$ can then be defined as follows.

Definition 3.5.1. *Let Q be a problem of type P(S,LC,BNS$_L$S$_N$,FV,RI) and $\bar{Q}$ be the linear relaxation of problem Q as defined in Section 3.4. Let $\phi_j : H_j \to \mathbb{R}$ be the lower semi-continuous concave objective function defined on variable j in problem Q, where H_j is the nonempty convex set in $\mathbb{R}$ defined as $H_j = \{x : l_j \leq x \leq u_j\}$. If $j \in B$ in the optimal solution to $\bar{Q}$,* *then the* concave underestimator *taken over H_j is the function $\hat{\phi}_j(x_j)$ such that*

$$\hat{\phi}_j(x_j) = \bar{\phi}_j(x_j) \tag{3.6}$$

If $j \in$ NL or $j \in$ NU in the optimal solution to $\bar{Q}$ then concave underestimator *taken over H_j is any function $\hat{\phi}_j(x_j)$ that satisfies the following properties:*

(OB.1) At each point $l_j \leq x_j \leq u_j$, $\hat{\phi}_j(x_j)$ is less than or equal to $\phi_j(x_j)$.

(OB.2) The function $\hat{\phi}_j(x_j)$ is concave for each $j \in J$.

(OB.3) If $j \in$ NL, the reduced cost function $\Delta_j(x_j)$ is monotonically increasing for $x_j \geq l_j$. If $j \in$ NU, the reduced cost function $\Delta_j(x_j)$ is monotonically decreasing for $x_j \leq u_j$.

(OB.4) If $j \in$ NL, then $\Delta_j(x_j) \geq 0$ for $x_j \geq l_j$. If $j \in$ NU, then $\Delta_j(x_j) \geq 0$ for $x_j \leq u_j$.

The concave underestimator relaxation of problem Q, denoted $\hat{Q}$, can now be defined as follows:

$$(\hat{Q}) \qquad \min \hat{\phi}(\underline{x}) \text{ s.t. } \underline{x} \in \hat{S} = \bar{X} \cap \hat{H} \tag{3.7}$$

where $\bar{X}$ is defined in equation (3.2), and $\hat{H} = \left\{\underline{x} : \hat{\underline{l}} \leq \underline{x} \leq \hat{\underline{u}}\right\}$ with $\hat{\underline{l}} \in \left(\ldots, \hat{l}_j, \ldots\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ and $\hat{\underline{u}} \in (\ldots, \hat{u}_j, \ldots)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ where

$$\hat{l}_j = \begin{cases} l_j & \text{if } j \in NL \\ -\infty & \text{if } j \in B \cup NU \end{cases} \tag{3.8}$$

$$\hat{u}_j = \begin{cases} u_j & \text{if } j \in NU \\ +\infty & \text{if } j \in B \cup NL \end{cases} \tag{3.9}$$

Problem $\hat{Q}$, the concave underestimator relaxation of problem Q, is generally referred to as the nonconvex relaxation of problem Q, for reasons that will become apparent subsequently in this Chapter.

Note that the specification of $\hat{l}_j$, $\hat{u}_j$, and $\hat{\phi}_j(x_j)$ means that problem $\hat{Q}$ exhibits three important properties. First, because $\hat{Q}$ satisfies conditions (R1) and (R2) given at the start of this Chapter, $\hat{Q}$ is a relaxation of subproblem Q (property (OB.1)). Hence, $\nu\left[\hat{Q}\right] \leq \nu[Q]$. Second, the nonconvex relaxation $\hat{Q}$ is purposely formulated so that its optimal solution corresponds to the optimal solution to the linear relaxation $\bar{Q}$. Thus $\underline{\hat{x}} = \underline{\bar{x}}$, where $\underline{\hat{x}} = (\ldots, \hat{x}_j, \ldots)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ is the optimal solution vector for problem $\hat{Q}$ (properties (OB.2), (OB.3), and (OB.4)). Finally, $\hat{l}_j$, $\hat{u}_j$, and $\hat{\phi}_j(x_j)$ have been defined in such a way that the parametric analysis of problem $\hat{Q}$ is particularly easy to perform.

Three possible specifications of $\hat{\phi}_j(x_j)$ for $j \in NL \cup NU$ are given in the remainder of this Section. The three specifications are called the *linear*, *concave*, and *mixed* formulations, and are identified by the superscripts L, C, and M respectively.

3.5.1 Linear Formulation

The objective function $\hat{\phi}_j^L(x_j)$ for the linear formulation when $j \in NL \cup NU$ is defined as the linear function

$$\hat{\phi}_j^L(x_j) = \bar{\phi}_j(x_j) \tag{3.10}$$

That is, $\hat{\phi}^L(\underline{x}) = \bar{\phi}(\underline{x})$, and it is trivial to show that the properties (OB.1) through (OB.4) hold.

3.5.2 Concave Formulation

In the second, concave, formulation, the specification of the objective function $\hat{\phi}_j^C(x_j)$ depends upon whether $j \in NL$ or $j \in NU$. If $j \in NL$, then

$$\hat{\phi}_j^C(x_j) = \begin{cases} \min\{\phi_j(x_j), \phi_j(u_j) + \bar{z}_j \cdot (x_j - u_j)\} & x_j \leq u_j \\ \phi_j(u_j) + \bar{z}_j \cdot (x_j - u_j) & x_j > u_j \end{cases} \tag{3.11}$$

where $\bar{z}_j$ is defined previously. If $j \in NU$, then

$$\hat{\phi}_j^C(x_j) = \begin{cases} \min\{\phi_j(x_j), \phi_j(l_j) + \bar{z}_j \cdot (x_j - l_j)\} & x_j \geq l_j \\ \phi_j(l_j) + \bar{z}_j \cdot (x_j - l_j) & x_j < l_j \end{cases} \tag{3.12}$$

Lemma 3.5.1. *Property (OB.1) holds for $\hat{\phi}_j^C(x_j)$ when $j \in$ NL and $j \in$ NU.*

Proof. When $l_j \leq x_j \leq u_j$ then $\hat{\phi}_j^C(x_j)$ is defined as the minimum of two (concave) functions, one of which is the function $\phi_j(x_j)$. Hence property (OB.1) trivially holds. □

Lemma 3.5.2. *Property (OB.2) holds for $\hat{\phi}_j^C(x_j)$ when $j \in$ NL and $j \in$ NU.*

Proof. The function $\hat{\phi}_j^C(x_j)$ is defined as the minimum of two concave functions, and is therefore itself concave. □

Lemma 3.5.3. *Property (OB.3) holds for $\hat{\phi}_j^C(x_j)$ when $j \in$ NL and $j \in$ NU.*

Proof. Consider the case where $j \in NL$. We need to show that $\Delta_j\left(x_j^1\right) \leq \Delta_j\left(x_j^2\right)$ where $l_j \leq x_j^1 \leq x_j^2$. First, if $\hat{\phi}_j^C\left(x_j^1\right) = \phi_j(u_j) + \bar{z}_j \cdot \left(x_j^1 - u_j\right)$ and $\hat{\phi}_j^C\left(x_j^2\right) = \phi_j(u_j) + \bar{z}_j \cdot \left(x_j^2 - u_j\right)$, then $\Delta_j\left(x_j^1\right) = \Delta_j\left(x_j^2\right)$ and property (OB.3) trivially holds. Second, consider the case where $\hat{\phi}_j^C\left(x_j^1\right) = \phi_j\left(x_j^1\right)$ and $\hat{\phi}_j^C\left(x_j^2\right) = \phi_j(u_j) + \bar{z}_j \cdot \left(x_j^2 - u_j\right)$. Because $\hat{\phi}_j^C\left(x_j^1\right) = \phi_j\left(x_j^1\right)$, then by definition $\hat{\phi}_j^C\left(x_j^1\right) \leq \phi_j(u_j) + \bar{z}_j \cdot \left(x_j^1 - u_j\right)$. Hence

$$\begin{aligned}
\Delta_j\left(x_j^1\right) &= \phi_j\left(x_j^1\right) - z_j x_j^1 \\
&\leq \phi_j(u_j) + \bar{z}_j \cdot \left(x_j^1 - u_j\right) - z_j x_j^1 \\
&= \phi_j(u_j) - z_j u_j \\
&= \Delta_j\left(x_j^2\right)
\end{aligned}$$

and property (OB.3) holds. Finally, consider the situation where both $\hat{\phi}_j^C\left(x_j^1\right) = \phi_j\left(x_j^1\right)$ and $\hat{\phi}_j^C\left(x_j^2\right) = \phi_j\left(x_j^2\right)$. Because $\phi_j(x_j)$ (and hence $\Delta_j(x_j)$) is concave, and

$$\begin{aligned}
\Delta_j(u_j) - \Delta_j(l_j) &= \bar{r}_j(u_j - l_j) \\
&\geq 0
\end{aligned}$$

and by definition

$$\Delta_j\left(x_j^1\right) \le \Delta_j\left(u_j\right)$$
$$\Delta_j\left(x_j^2\right) \le \Delta_j\left(u_j\right)$$

then $\Delta_j\left(x_j^2\right) \ge \Delta_j\left(x_j^1\right)$ and property (OB.3) holds. Therefore, property (OB.3) holds when $j \in NL$. The proof of property (OB.3) is similar for the case where $j \in NU$. □

Lemma 3.5.4. *Property (OB.4) holds for $\hat{\phi}_j^C(x_j)$ when $j \in \mathrm{NL}$ and $j \in \mathrm{NU}$.*

Proof. Note that $\hat{\phi}_j^C(l_j) = \bar{\phi}_j^L(l_j)$, and hence $\Delta_j(l_j)$ is the same for both functions. Combined with property (OB.3) and the fact that property (OB.4) holds for $\hat{\phi}_j^L(x_j)$, this implies property (OB.4) when $j \in NL$. The proof is similar for the case where $j \in NU$. □

3.5.3 Mixed Formulation

In the third, mixed, formulation, the objective function $\hat{\phi}_j^M(x_j)$ is specified simply as

$$\hat{\phi}_j^M(x_j) = \max\left\{\hat{\phi}_j^L(x_j), \hat{\phi}_j^C(x_j)\right\} \tag{3.13}$$

In general, $\hat{\phi}_j^M(x_j)$ is neither concave nor convex (hence problem $\hat{Q}$ is referred to as the nonconvex relaxation of Q). However, note that for any particular variable j, $\hat{\phi}_j^M(x_j)$ can equivalently be specified as *either* $\hat{\phi}_j^L(x_j)$ *or* $\hat{\phi}_j^C(x_j)$, whichever is greater. Thus, since both $\hat{\phi}_j^L(x_j)$ and $\hat{\phi}_j^C(x_j)$ are concave underestimators for $\phi_j(x_j)$, then $\hat{\phi}_j^M(x_j)$ can be treated in the

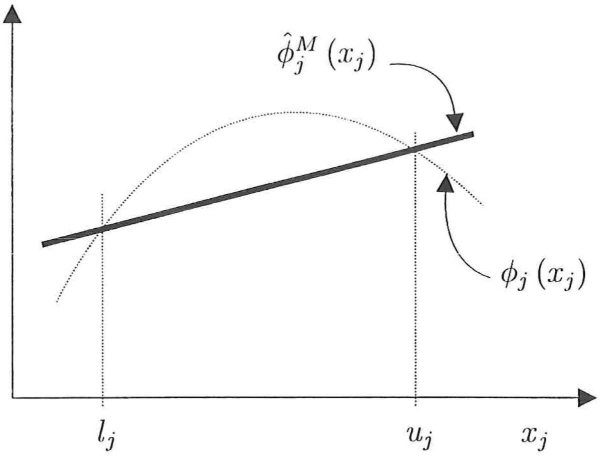

Figure 3.4: Arc cost function $\hat{\phi}_j^M(x_j)$ for $j \in B$

same manner as a concave underestimator for $\phi_j(x_j)$ even though it is not necessarily a concave function.

Figures 3.4, 3.5 and 3.6 show typical representations of $\hat{\phi}_j^M(x_j)$ when $j \in B$, $j \in NL$ and $j \in NU$, respectively.

3.6 Post-Optimal Parametric Analysis

The parametric analysis given below describes how the solution to the non-convex relaxation $\hat{Q}$ changes as the value of a single variable varies from its optimal value in problem $\hat{Q}$. Let $k \in J$ be the index of the arc being altered, and let "$\hat{Q} \mid x_k = \hat{x}_k + \delta_k$" denote problem $\hat{Q}$ augmented with the constraint "$x_k = \hat{x}_k + \delta_k$" where $\hat{x}_j = \bar{x}_j$ and δ_k is a (as yet unspecified) scalar. To describe the parametric analysis of problem "$\hat{Q} \mid x_k = \hat{x}_k + \delta_k$" using δ_k as the parameter, it is convenient to define the *parametric function* $\theta_k(\delta_k)$ as

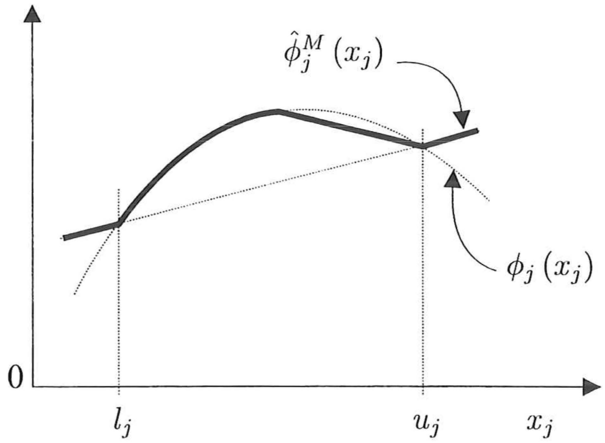

Figure 3.5: Arc cost function $\hat{\phi}_j^M(x_j)$ for $j \in NL$

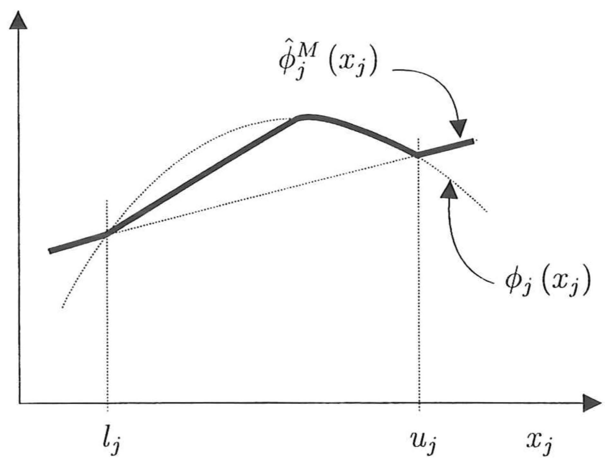

Figure 3.6: Arc cost function $\hat{\phi}_j^M(x_j)$ for $j \in NU$

$$\theta_k(\delta_k) = \nu\left[\hat{Q} \mid x_k = \hat{x}_k + \delta_k\right] - \nu\left[\hat{Q}\right] \tag{3.14}$$

Note that $\theta_k(\delta_k)$ is a unimodal function with a minimum value of $\theta_k(\delta_k) = 0$ at $\delta_k = 0$. In addition, $\theta_k(\delta_k)$ is nonincreasing for all $\delta_k \leq 0$ and nondecreasing for all $\delta_k \geq 0$.

To express $\theta_k(\delta_k)$ explicitly, it is useful to refer to the reduced cost function, $\Delta_j(x_j)$, associated with $\hat{\phi}_j(x_j)$. The calculation of the parametric function $\theta_k(\delta_k)$ depends upon whether $k \in B$, $k \in NL$, or $k \in NU$.

If $k \in NL$, then changing x_k from $\hat{x}_k$ to $\hat{x}_k + \delta_k$ in problem $\hat{Q}$ means that the minimum objective function value will increase by $\Delta_k(l_k + \delta_k) - \Delta_k(l_k)$ if $\delta_k \geq 0$, and by an infinite amount if $\delta_k < 0$. That is, if $k \in NL$, then

$$\theta_k(\delta_k) = \begin{cases} +\infty & \text{if } \delta_k < 0 \\ \Delta_k(l_k + \delta_k) - \Delta_k(l_k) & \text{if } \delta_k \geq 0 \end{cases} \tag{3.15}$$

Similarly, if $k \in NU$, then

$$\theta_k(\delta_k) = \begin{cases} \Delta_k(u_k + \delta_k) - \Delta_k(u_k) & \text{if } \delta_k \leq 0 \\ +\infty & \text{if } \delta_k > 0 \end{cases} \tag{3.16}$$

If $k \in B$, then changing x_k from $\hat{x}_k$ to $\hat{x}_k + \delta_k$ in problem $\hat{Q}$ means that a *single* nonbasic variable x_j will change from $\hat{x}_j$ to $\hat{x}_j - (\delta_k/\bar{a}_{kj})$; and the minimum objective function value will increase by $\Delta_j(\hat{x}_j - (\delta_k/\bar{a}_{kj})) - \Delta_j(\hat{x}_j)$. The reason why the value of only a single nonbasic variable will change in this case is as follows. First, $\hat{l}_j$ and $\hat{u}_j$ for each $j \in N$ are defined such that these capacities will not be binding for any value of $\delta_k \neq 0$. Second, for any given δ_k, each $\hat{\phi}_j\left(\hat{x}_j - (\delta_k/\bar{a}_{kj})\right)$ can be treated as a single concave function. Combined, these properties mean that $\Delta_j(x_j)$ can be treated as an uncapacitated concave function.

To describe the calculation of $\theta_k(\delta_k)$ when $k \in B$, we define another function, denoted $\theta_{kj}(\delta_k)$, as

$$\theta_{kj}(\delta_k) = \begin{cases} \Delta_j\left(\hat{x}_j - (\delta_k/\bar{a}_{kj})\right) - \Delta_j(\hat{x}_j) & \text{if } \delta_k < 0 \text{ and } j \in J_k^- \\ 0 & \text{if } \delta_k = 0 \\ \Delta_j\left(\hat{x}_j - (\delta_k/\bar{a}_{kj})\right) - \Delta_j(\hat{x}_j) & \text{if } \delta_k > 0 \text{ and } j \in J_k^+ \\ +\infty & \text{otherwise} \end{cases} \tag{3.17}$$

where the index subsets J_k^+ and J_k^- are defined as

$$J_k^- = \{j : (j \in NL \text{ and } \bar{a}_{kj} > 0) \text{ or } (j \in NU \text{ and } \bar{a}_{kj} < 0)\} \tag{3.18}$$

$$J_k^+ = \{j : (j \in NL \text{ and } \bar{a}_{kj} < 0) \text{ or } (j \in NU \text{ and } \bar{a}_{kj} > 0)\} \tag{3.19}$$

Then, if $k \in B$, the function $\theta_k(\delta_k)$ is given by

$$\theta_k(\delta_k) = \min_{j \in J} \{\theta_{kj}(\delta_k)\} \tag{3.20}$$

Figure 3.7 shows a typical representation of $\theta_k(\delta_k)$ when $k \in B$.

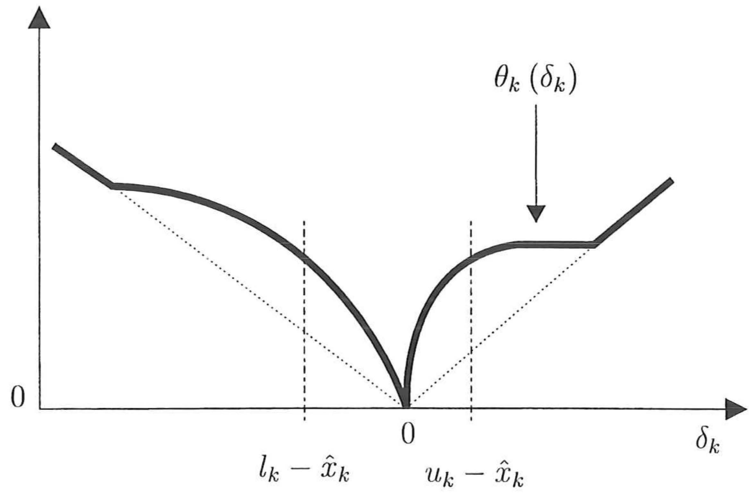

Figure 3.7: Parametric function $\theta_k(\delta_k)$ for $k \in B$

Notice that, for any $k \in J$, the function $\theta_k(\delta_k)$ given in (3.15), (3.16), and (3.20) is based directly on information available in the solution to the linear program $\bar{Q}$. The parametric analysis of the nonconvex relaxation can then be used to obtain information about the solution to the original problem Q. The following Chapter of this thesis develops a solution algorithm for problems of type P(S,LC,BN$S_L S_N$,FV,RI) that uses the concave underestimator analysis presented here.

Chapter 4

Enhanced Capacity Improvement

4.1 Introduction

In this chapter, we consider the solution of mixed–integer concave cost network flow problems of type P[S,LC,NBS_L,FV,RI]. Throughout this chapter we denote problem P as a generic instance of a problem of type P[S,LC,NBS_L,FV,RI]. Problem P can be formulated as follows:

$$(P) \qquad \min \phi\left(\underline{x}\right) \text{ s.t. } \left(\underline{x}\right) \in S, \underline{x} = \left(\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}\right) \in \mathbb{Y}^{n+m}$$

where the domain $\mathbb{Y}^{n+m}$ is defined to be the set $\mathbb{Y}^{n+m} = \{\mathbb{R}^n; \mathbb{Z}^m\}$, $\underline{x} = (\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}) = (\ldots, x_j, \ldots)^{\mathrm{T}} \in \mathbb{Y}^{n+m}$ is the vector of decision variables with index set $J = \{1, \ldots, j, \ldots, n+m\}$, $\underline{x}_{\mathbb{R}} \in \mathbb{R}^n$ is the vector of continuous real valued solution variables, $\underline{x}_{\mathbb{Z}} \in \mathbb{Z}^m$ is the vector of integer valued solution variables, S is the set defined as $S = X \cap H$ where

$$X = \left\{ \underline{x} \in \mathbb{R}^{n+m} : \left\{ \begin{array}{ll} A\underline{x}_{\mathbb{R}} & = \underline{b} \\ \\ C\underline{x} & = \underline{g} \end{array} \right\} \right\} \tag{4.1}$$

and H is the hyperrectangle representing the flow bounds given by

$$H = \left\{ \underline{x} \in \mathbb{R}^{n+m} : \underline{l} \leq \underline{x} \leq \underline{u} \right\} \tag{4.2}$$

Finally, $\phi_j\left(x_j\right)$ is the real–valued concave separable objective function for arc j, and $\phi\left(\underline{x}\right) = \sum_{j \in J} \phi_j\left(x_j\right)$.

The focus of this chapter is to illustrate the application of the concave underestimator analysis of the previous chapter to a solution algorithm. To this end, a branch and bound algorithm for mixed–integer concave minimisation of problems of type P is developed that incorporates concave underestimator analysis. The resulting algorithm is an extension of the *capacity improvement* algorithms presented in the literature. Capacity improvement is a domain reduction technique wherein the simple variable bounds are systematically tightened, reducing the size of the hyperrectangle H. Capacity improvement techniques for the solution of continuous problems of type P (i.e. problems for which $m = 0$) based on a *linear* relaxation have been studied by Lamar, Sheffi & Powell (1990), Thakur (1990), Lamar (1993*a*), Lamar (1995), Ryoo & Sahinidis (1996), Shectman & Sahinidis (1998). In this chapter, we extend and generalise capacity improvement by using the concave underestimator analysis of the previous chapter. The capacity improvement bounds thus produced can be considerably tighter than those produced by the traditional linear relaxation approach.

The remainder of this Chapter is organised as follows. First, the generic rectangular branch and bound solution algorithm for problems of type P is presented. Node selection, bounding and partitioning rules for rectangular branch and bound algorithms common in the literature are then presented, followed by a discussion of convergence issues. Second, the theory of capacity improvement utilising the concave underestimator analysis of the previous Chapter is developed. Finally, an "enhanced" branch and bound algorithm incorporating capacity improvement is presented, and convergence issues discussed.

4.2 Rectangular Branch and Bound

The standard rectangular branch and bound algorithm for mixed integer separable concave minimisation problems of type P presented here is a straightforward combination of the branch and bound algorithms for the continuous and integer cases reported in the literature. Branch and bound procedures solve problems of type P by partitioning the feasible region into successively smaller subregions. Lower bounds to the optimal objective function value of the problem over these subregions, and upper bounds to the optimal objective function value of the original problem, are then determined. Regions that have a lower bound greater than the incumbent (i.e. best known) upper bound to the optimal solution value of the original problem are removed from consideration, as they cannot contain a solution better than the incumbent. The remaining regions are then partitioned still further, and the process repeats until either the difference between the minimum of the lower bounds and the incumbent upper bound is smaller than some predetermined tolerance or there are no unfathomed subregions remaining.

The rectangular branch and bound algorithm begins with hyperrectangle $H = \{\underline{x} \in \mathbb{R}^{n+m} : \underline{l} \leq \underline{x} \leq \underline{u}\} \in \mathbb{R}^{n+m}$ as formed by the variable upper and lower bounds from problem P and defined in equation (4.2). A *m–integral rectangular partition* of hyperrectangle H is defined as follows:

Definition 4.2.1. *A set $\{H^i : i \in I\}$ of hyperrectangles is said to be an* m-integral rectangular partition *of hyperrectangle H if*

$$H \supseteq \bigcup_{i \in I} H^i$$
$$H \cap \mathbb{Y}^{n+m} = \bigcup_{i \in I} H^i \cap \mathbb{Y}^{n+m}$$
$$\Im H^i \cap \Im H^j = \emptyset \qquad \forall i, j \in I, \qquad i \neq j$$

where $\Im H^i$ denotes the the interior of hyperrectangle H_i.

We now assume an m–integral rectangular partition of the variable bound hyperrectangle H defined in equation (4.2) exists, and define hyperrectangle $H_Q \in \{H^i : i \in I\}$ as the partition element of H currently under consideration. Problem Q is then defined as the subproblem of P associated with H_Q. Specifically:

$$(Q) \qquad \min \phi\left(\underline{x}\right) \text{ s.t. } \left(\underline{x}\right) \in S = X \cap H_Q, \underline{x} = \left(\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}\right) \in \mathbb{Y}^{n+m}$$

where $H_Q = \left\{\underline{x} : \underline{l}_Q \leq \underline{x} \leq \underline{u}_Q\right\}$ with $\underline{l}_Q = \left(\ldots, l_j^Q, \ldots\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ and $\underline{u}_Q = \left(\ldots, u_j^Q, \ldots\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$, and $H_Q \subseteq H$.

Rather than attempting to solve Q directly, we instead solve a linear objective and feasibility relaxation of Q. We let $\bar{Q}$ denote this relaxation of problem Q. That is,

$$(\bar{Q}) \qquad \min \bar{\phi}_Q(\underline{x}) \quad \text{s.t.} \quad \underline{x} \in S_Q = X \cap H_Q,\ \underline{x} \in \mathbb{R}^{n+m}$$

where $\bar{\phi}_Q(\underline{x})$ is the convex envelope of $\phi(\underline{x})$ on H_Q. Because of the separability of $\phi(\underline{x})$, we have $\bar{\phi}_Q(\underline{x}) = \sum_{j\in J} \bar{\phi}_{Qj}(x_j)$ where $\bar{\phi}_{Qj}(x_j)$ is the convex envelope of $\phi_{Qj}(x_j)$ over the range $l_{Qj} \leq x_j \leq u_{Qj}$. Since $\bar{Q}$ is a relaxation of Q, $\nu\left[\bar{Q}\right]$ is a valid lower bound to $\nu[Q]$. Specifically, we define the *relaxation lower bound* as

$$lb[Q] = \nu\left[\bar{Q}\right] \tag{4.3}$$

Problem Q and its descendants can be removed from further consideration if the following *fathoming* criterion, denoted criterion (F), is satisfied:

$$(F) \qquad lb[Q] \geq ub[P]$$

In the traditional branch and bound, the value of $ub[P]$ is taken as the objective function value of the "incumbent" solution[1] to P, and the value for $lb[Q]$ is given by the relaxation lower bound defined in equation (4.3). If the fathoming criterion (F) is satisfied, then no further evaluation of problem Q is required and another subproblem of P can be selected and evaluated as the current subproblem. If, however, the fathoming criterion

[1]That is, the feasible solution to P with the smallest objective value of those feasible solutions found so far.

is not satisfied, then one of two "branching actions" defined below must be taken.

(B1) - Partitioning "Separate" at the current subproblem by partitioning the hyperrectangle H_Q into η smaller hyperrectangles, thereby replacing Q with η new subproblems of P;

(B2) - Persisting "Continue" at the current subproblem by evaluating a tighter relaxation of Q than that provided by $\bar{Q}$.

Following the partitioning action (*B1*), a new subproblem will be selected for evaluation by the branch and bound procedure. Following the persisting action (*B2*), the branch and bound procedure will attempt to produce a tighter value of $lb\,[Q]$ in order to fathom problem Q. In the traditional branch and bound procedure, however, only partitioning action (*B1*) is considered. Once all the subproblems of P have been fathomed, the branch and bound procedure terminates, and the current incumbent solution to P is identified as the optimal solution to P.

4.2.1 Formal Algorithm Statement

Formally, the branch and bound algorithm presented in the previous Section can be described as follows.

Algorithm 4.2.1. Branch and Bound for Minimisation of problems of type P[S,LC,NBS_L,FV,RI]

Step 0.1 *Define the set* FEAS $\subset S$ *as the set of* known *feasible solutions to problem P. Note that the set* FEAS *may be empty.*

Step 0.2 *Set*

$$ub[P] = \begin{cases} min\{\phi(\underline{x}) : \underline{x} \in \text{FEAS}\} & if\ \text{FEAS} \neq \emptyset \\ \infty & if\ \text{FEAS} = \emptyset \end{cases}$$

Step 0.3 *Define the* incumbent solution vector $\tilde{\underline{x}}$ *as the vector* $\underline{x} \in$ FEAS *satisfying* $\phi(\underline{x}) = ub[P]$

Step 0.4 *Define the set* SUB *of active subproblems to problem* P *as* SUB $= \{P\}$.

Step 0.5 *Set the iteration counter* $i \leftarrow 1$

Step i.1 *Select a subproblem* $Q \in$ SUB *as the current subproblem. This step is known as "branching" or "node selection".*

Step i.2 *Construct an m–integral partition of* H_Q *into finitely many subsets* $H_{Q^1}, \ldots, H_{Q^r}, \ldots, H_{Q^\eta}$, *and form the associated subproblems of* P *denoted* $Q^1, \ldots, Q^r, \ldots, Q^\eta$.

Step i.3 *Obtain* $lb[Q^r]$, *a lower bound to* $\nu[Q^r]$ *for each* Q^r. *This step is known as "bounding". Define* $\overline{\underline{x}}^r$ *as the solution vector* $\underline{x} \in S_{Q^r}$ *satisfying* $\bar{\phi}(\underline{x}) = lb[Q^r]$.

Step i.4 *For each* Q^r, *if* $\overline{\underline{x}}^r = (\overline{\underline{x}}^r_{\mathbb{R}}, \overline{\underline{x}}^r_{\mathbb{Z}})$ *such that* $\overline{\underline{x}}^r_{\mathbb{R}} \in \mathbb{R}^n$ *and* $\overline{\underline{x}}^r_{\mathbb{Z}} \in \mathbb{Z}^m$, *then update* FEAS *by setting* FEAS $\leftarrow$ FEAS $\cup\, \overline{\underline{x}}^r$.

Step i.5 *Update* $ub[P] = min\{\phi(\underline{x}) : \underline{x} \in \text{FEAS}\}$ *and define the incumbent solution vector* $\tilde{\underline{x}}$ *as the vector* $\underline{x} \in$ FEAS *satisfying* $\phi(\underline{x}) = ub[P]$.

Step i.6 *Delete (fathom) all subproblems* $Q \in \mathrm{SUB}$ *satisfying* $lb\,[Q] \geq ub\,[P]$.

Step i.7 *If* $\mathrm{SUB} \neq \emptyset$ *set the iteration counter* $i \leftarrow i+1$ *and go to step* i.*1. Otherwise,* $\nu\,[P] \geq ub\,[P]$ *and the current incumbent solution is optimal. Define* $\underline{x}^* = \tilde{\underline{x}}$ *as the optimal solution vector for problem* P *and exit the solution procedure.*

4.2.1.1 Node Selection

Step *i*.1 of the branch and bound algorithm, the selection of an active subproblem for further refinement, is called *node selection.* The most common node selection rule presented in the literature for general branch and bound algorithms is *lowest-bound-first-search.*

Definition 4.2.2. Lowest–Bound–First–Search Branching Rule

Select the subproblem $Q \in \mathrm{SUB}$ *that has the least lower bound associated with it; that is, select the subproblem* $Q \in \mathrm{SUB}$ *satisfying* $lb\,[Q] = \min\{lb\,[Q] : Q \in \mathrm{SUB}\}$

A simple node selection rule that is easier to implement than lowest–bound–first–search is *depth–first–search.*

Definition 4.2.3. Depth–First–Search Branching Rule

Select the subproblem $Q \in \mathrm{SUB}$ *most recently created. That is, the set SUB is treated as a LIFO (Last–In–First–Out) stack.*

4.2.1.2 Bounding

Step $i.3$ of the branch and bound algorithm requires the calculation of $lb\,[Q]$, a lower bound on the optimal solution to problem Q. The calculation of $lb\,[Q]$ is relatively straight forward in the standard branch and bound procedure. Recall the definition of problem $\bar{Q}$ as follows:

$$(\bar{Q}) \qquad \min \bar{\phi}_Q\left(\underline{x}\right) \text{ s.t. } \underline{x} \in S_Q = X \cap H_Q,\ \underline{x} \in \mathbb{R}^{n+m}$$

where $\bar{\phi}_Q\left(\underline{x}\right) = \sum_{j \in J} \bar{\phi}_{Qj}\left(x_j\right)$ is the convex envelope of $\phi\left(\underline{x}\right) = \sum_{j \in J} \phi_j\left(x_j\right)$ on H_Q. That is:

$$\bar{\phi}_{Qj}\left(x_j\right) = f_{Qj} + c_{Qj} \times x_j \tag{4.4}$$

where

$$c_{Qj} = \frac{\phi_j\left(u_j^Q\right) - \phi_j\left(l_j^Q\right)}{u_j^Q - l_j^Q} \tag{4.5}$$

$$f_{Qj} = \phi_j\left(l_j^Q\right) - c_{Qj} \times l_j^Q \tag{4.6}$$

Problem $\bar{Q}$ is a linear program, and is therefore relatively easy to solve. Since problem $\bar{Q}$ is a relaxation of problem Q, we have

$$lb\,[Q] = \nu\left[\bar{Q}\right] \leq \nu\,[Q] \tag{4.7}$$

That is, the optimal objective function value for problem $\bar{Q}$ forms a valid lower bound to the optimal objective function value for problem Q.

4.2.1.3 Partitioning Schemes

The rectangular partitioning schemes presented here are variants on the weak partitioning rule first proposed by Falk & Soland (1969) and are common in the literature (see, for example, Horst et al. (1995) and Benson (1996)). Weak partitioning consists of two steps; first the "branching variable" must be selected, and second the hyperrectangle under consideration is partitioned at some value of the branching variable.

Two variable selection strategies common in the literature for the case in which all variables are continuous (that is, $\underline{x} \in \mathbb{R}^n$) are *maximum-gap* and *longest-edge* variable selection.

Definition 4.2.4. Maximum-gap variable selection

Define the set J' to be the index set of arcs $j \in J$ for which $\phi_j(\bar{x}_{Qj}) - \bar{\phi}_{Qj}(\bar{x}_{Qj}) > 0$. Select a variable $k \in J'$ satisfying

$$\phi_k(\bar{x}_{Qk}) - \bar{\phi}_{Qk}(\bar{x}_{Qk}) = \max_{j \in J'} \left\{ \phi_j(\bar{x}_{Qj}) - \bar{\phi}_{Qj}(\bar{x}_{Qj}) \right\}$$

where $\bar{x}_{Qj}$ is the j*-th element of $\underline{\bar{x}}_Q$, the optimal solution vector for problem $\bar{Q}$, the linear programming relaxation of problem Q.*

Definition 4.2.5. Longest-edge variable selection

Define the set J' to be the index set of arcs $j \in J$ for which $\phi_j(\bar{x}_{Qj}) - \bar{\phi}_{Qj}(\bar{x}_{Qj}) > 0$. Select a variable $k \in J'$ satisfying

$$u_{Qk} - l_{Qk} = \max_{j \in J'} \{u_{Qj} - l_{Qj}\}$$

Once the branching variable $k \in J$ has been selected, two hyperrectangles H_{Q^1} and H_{Q^2} are constructed as follows:

$$H_{Q^1} = \{\underline{x} \in H_Q : x_k \leq \omega_{Qk}\}$$

$$H_{Q^2} = \{\underline{x} \in H_Q : x_k \geq \omega_{Qk}\}$$

where $l_k < \omega_{Qk} < u_k$. The value of ω_{Qk} can be determined using *bisection* (Horst & Tuy 1996), $\bar{x}$ *subdivision* (Horst & Tuy 1996), or *incumbent subdivision* (see, for example, Ryoo & Sahinidis (1996)).

Definition 4.2.6. Bisection

Set $\omega_{Qk} = (u_{Qk} - l_{Qk})/2$

Definition 4.2.7. $\bar{x}$ subdivision

Set $\omega_{Qk} = \bar{x}_{Qk}$

Definition 4.2.8. Incumbent subdivision

If $\tilde{\underline{x}}$*, the solution vector of the current incumbent solution, is contained in* H_Q*, then set* $\omega_{Qk} = \tilde{x}_k$ *where* $\tilde{x}_k$ *is the k-th element of* $\tilde{\underline{x}}$*.*

The explicit branching scheme used can then be constructed using the branching variable selection and branching point selection elements. Two common schemes are longest-edge with bisection, and maximum-gap with $\bar{x}$ subdivision (see, for example, Benson (1995) and Horst & Tuy (1996)). Alternative schemes include the one presented in Ryoo & Sahinidis (1996), which utilises longest–edge variable selection and bisection if the iteration count is some multiple of a predefined integer value. Otherwise incumbent subdivision is used if $\tilde{\underline{x}} \in H_Q$, or else $\bar{x}$ subdivision is used if $\tilde{\underline{x}} \notin H_Q$.

A partitioning scheme for the continuous case may be logically extended to the general mixed integer case as follows. The branching variable $k \in J$

is chosen using a continuous branching variable selection scheme (such as the longest-edge or maximum-gap methods) modified as follows:

Definition 4.2.9. Mixed integer modified variable selection

Define the set J' to be the index set of arcs $j \in J$ for which $\phi_j(\bar{x}_{Qj}) - \bar{\phi}_{Qj}(\bar{x}_{Qj}) > 0$. If $J' \neq \emptyset$, select a variable $k \in J'$ using the continuous branching variable scheme. Otherwise, select a variable $k \in J$ such that $k > n$ (that is, $x_k \in \mathbb{Z}$ in any feasible solution to problem Q) and $\bar{x}_k \notin \mathbb{Z}$.

The preliminary branching point ω_{Qk} is selected using a branching value selection method such as the bisection, $\bar{x}$ subdivision, or incumbent subdivision rules defined above. Then, if $k \leq n$ (that is, x_k may be real-valued), hyperrectangles H_{Q^1} and H_{Q^2} are constructed as previously. That is

$$H_{Q^1} = \{\underline{x} \in H_Q : x_k \leq \omega_{Qk}\}$$

$$H_{Q^2} = \{\underline{x} \in H_Q : x_k \geq \omega_{Qk}\}$$

Otherwise, $k > n$, and thus x_k must be integer valued in the optimal solution to Q. Construct hyperrectangles H_{Q^1} and H_{Q^2} such that

$$H_{Q^1} = \{\underline{x} \in H_Q : x_k \leq \lfloor \omega_{Qk} \rfloor\}$$

$$H_{Q^2} = \{\underline{x} \in H_Q : x_k \geq \lfloor \omega_{Qk} \rfloor + 1\}$$

where $\lfloor t \rfloor$ is the largest integer such that $\lfloor t \rfloor \leq t$.

4.2.2 Convergence of the Algorithm

From Horst & Tuy (1996) we have the following two definitions.

Definition 4.2.10. *A selection operation is said to be* bound improving *if, at least each time after a finite number of steps, the partition element $H_{\underline{Q}}$ of H associated with subproblem $\underline{Q}$ such that*

$$lb[\underline{Q}] = \min_{Q \in SUB} (lb[Q])$$

is selected for further refinement.

An example of a bounding improving selection operation is the *lowest-bound-first-search* scheme discussed in Section 4.2.1.1. Note however that the *depth-first-search* procedure is not bound improving.

Definition 4.2.11. *A combined partitioning and bounding operation is called* consistent *if at every step any unfathomed partition element can be further refined, and if any infinitely decreasing sequence $\{S_{Q^r}\}$ of successively refined partition elements such that $\{S_{Q^{r+1}}\} \subset \{S_{Q^r}\}$ satisfies*

$$\lim_{r \to \infty} (ub[Q^r] - lb[Q^r]) = 0$$

Longest-edge with bisection, maximum–gap with bisection, and maximum-gap with $\bar{x}$ subdivision are common partitioning schemes that, when combined with the bounding operation from Section 4.2.1.2, are consistent for the case where all variables are continuous (see, for example, Horst et al. (1995) and Horst & Tuy (1996)). The following theorem is a logical extension of the definition of consistency.

Theorem 4.2.1. *Assume a combined partitioning and bounding operation for a continuous variable branch and bound procedure is consistent. Then* *the mixed-integer modification of that partitioning scheme, as defined in Section 4.2.1.3, combined with the bounding operation is also consistent.*

Proof. Assume there exists an infinitely decreasing sequence $\{S_{Q^r}\}$ of successively refined partition elements such that $\{S_{Q^{r+1}}\} \subset \{S_{Q^r}\}$. Let $j_r \in J$ index the branching variable associated with S_{Q^r}. If $j_r > n$ (that is, variable x_{jr} is integer valued), then it must occur a finite number of times in the sequence. Removing the partition elements associated with each $j_r \in J$ for which $j_r > n$ from the sequence $\{S_{Q^r}\}$ leaves an infinitely decreasing subsequence of successively refined partition elements $\left\{S_{Q^{r'}}\right\}$ with associated branching variables indexed by $j_{r'} \in J$ such that $j_{r'} \leq n$. This sequence is generated by the assumed consistent combined partitioning and bounding operation for a continuous variable branch and bound procedure. Thus

$$\lim_{r' \to \infty} \left(ub[Q^{r'}] - lb[Q^{r'}]\right) = 0$$

and therefore

$$\lim_{r \to \infty} (ub[Q^r] - lb[Q^r]) = 0$$

□

Any branch and bound algorithm, such as the general algorithm presented here, will either terminate in a finite number of iterations, or it will not terminate in a finite number of iterations. The former is a called a finite branch and bound procedure, the later an infinite branch and bound procedure. Horst & Tuy (1996) prove the following convergence theorem for infinite branch and bound algorithms:

Theorem 4.2.2. *In an infinite branch and bound procedure, suppose that the partitioning and bounding operation is consistent and the selection operation is bound improving. Then the procedure is convergent.*

Thus, in general the branch and bound algorithm presented here will converge to the optimal solution of a problem of type P, provided of course the bounding operation is consistent and the selection operation is bound improving.

An excellent analysis of the finiteness property of branch and bound algorithms for minimisation of separable concave functions over a continuous polytope is given in Shectman & Sahinidis (1998). They develop two conditions that, if met, guarantee that a branch and bound algorithm based on a rectangular partition scheme and an LP-based bounding operation for problems of this type terminate finitely. These two conditions are as follows.

Condition 4.2.1. *For all nested sequences $\{S_{Q^r}\}$ of subdomains S_{Q^r} generated by the partitioning rule,*

$$\lim_{r \to \infty} \max_{j \in J} \left(u_j^{Q^r} - l_j^{Q^r} \right) = 0 \tag{4.8}$$

Condition 4.2.2. *If a subproblem contains a global solution point, the algorithm will construct a partition of the subproblem through that point.*

Branch and bound using solely a maximum gap variable selection and $\bar{x}$ subdivision partitioning scheme (such as the "relaxed" algorithm of Falk & Soland (1969)) fails condition (4.2.1). However, this scheme can be modified by implementing longest–edge variable selection followed by a bisection subdivision at every N iterations, where N is some prespecified positive scalar. At every other iteration, maximum gap variable selection and $\bar{x}$ subdivision are used. Such a modified scheme clearly meets condition (4.2.1) (Shectman & Sahinidis 1998). Other possible partitioning strategies developed in Shectman & Sahinidis (1998) include a similar scheme in which at every N iterations longest–edge variable selection is used; otherwise maximum gap variable selection is employed. Then, if the current incumbent solution lies within the hyperrectangle H_Q, incumbent subdivision is used to partition H_Q; otherwise a standard bisection subdivision is implemented.

Further, note that a finite branch and bound algorithm can easily be constructed for certain subclasses of the problems considered in this chapter. For example, for minimum concave cost pure network flow problems, it is well known that each extreme point solution is integer valued (Charnes & Cooper 1961). Hence, a branch and bound algorithm with a partitioning strategy that partitions the feasible region at integer flow values will be finite, since there are a finite number of integer values in the feasible region, and hence a finite number of possible partitions. Similarly, for separable concave minimisation problems when all variables are defined to be integer, a branch and bound algorithm implementing a partitioning strategy that partitions at integer variable values will also be finite. Finally, for those problems with piecewise linear and/or fixed charge objectives, a partitioning strategy that partitions at piecewise linear breakpoints will also be finite, since there are a finite number of such breakpoints, and hence a finite number of possible subdivisions.

4.3 Enhanced Branch-and-Bound Procedure

The branch and bound procedure described in the previous Section may be "enhanced" via *capacity improvement.* Capacity improvement attempts to obtain a tighter value of $lb\,[Q]$ than that given by the relaxation lower bound (equation (4.3)) by implementing the alternative branching action ($B2$).

4.3.1 Capacity Improvement

"Capacity Improvement" is one method of taking action ($B2$) (i.e. "persisting") at the current subproblem Q. In order to explain the concept of capacity improvement, we define another problem, denoted $\tilde{Q}$, whose feasible region, $\tilde{S}_Q$, is restricted to the (possibly empty) subset of S_Q in which the value of $\phi\,(\underline{x})$ is less than or equal to $ub\,[P]$. That is,

$$\left(\tilde{Q}\right) \qquad \min \phi\,(\underline{x}) \ \text{ s.t. } \ \underline{x} \in \tilde{S}_Q = X \cap H_Q \cap \{\underline{x} : \phi\,(\underline{x}) \leq ub\,[P]\}$$
$$\underline{x} = (\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}) \in \mathbb{Y}^{n+m} \tag{4.9}$$

Note that if $ub\,[P] \geq \nu\,[Q]$, then $\nu\left[\tilde{Q}\right] = \nu\,[Q]$. Otherwise, if $ub\,[P] < \nu\,[Q]$, then $\tilde{S}_Q$ is empty, and $\tilde{Q}$ is infeasible.

In addition, let Q^t for $t = 0, 1, 2, \ldots$ be a family of successively tighter *feasibility* relaxations of $\tilde{Q}$. Specifically,

$$(Q^t) \qquad \min \phi\left(\underline{x}\right) \text{ s.t. } \underline{x} \in S_Q^t = X \cap H_Q^t, \underline{x} = \left(\underline{x}_{\mathbb{R}}, \underline{x}_{\mathbb{Z}}\right) \in \mathbb{Y}^{n+m} \tag{4.10}$$

where $H_Q^t = \left\{\underline{x} : \underline{l}_Q^t \leq \underline{x} \leq \underline{u}_Q^t\right\}$ with $\underline{l}_Q^t = \left(\ldots, l_j^{Qt}, \ldots\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ and $\underline{u}_Q^t = \left(\ldots, u_j^{Qt}, \ldots\right)^{\mathrm{T}} \in \mathbb{R}^{n+m}$ such that for each t we have $S_Q^t \supseteq S_Q^{t+1} \supseteq \tilde{S}_Q$. For $t = 0$, we set $H_Q^0 = H_Q$ so that $S_Q^0 = S_Q$ and problem Q^0 is the same as problem Q. Note that, if $ub\left[P\right] \geq \nu\left[Q\right]$, then for each t we have $\nu\left[Q^t\right] = \nu\left[Q^{t+1}\right] = \nu\left[\tilde{Q}\right] = \nu\left[Q\right]$.

Next, as in the standard branch and bound algorithm presented previously, we define problem $\bar{Q}^t$ to be a linear programming relaxation of problem Q^t. Specifically,

$$(\bar{Q}^t) \qquad \min \bar{\phi}_Q^t\left(\underline{x}\right) \text{ s.t. } \underline{x} \in S_Q^t = X \cap H_Q^t \tag{4.11}$$

where $\bar{\phi}_Q^t\left(\underline{x}\right) = \sum_{j \in J} \bar{\phi}_{Qj}^t\left(x_j\right)$ is the convex envelope of $\phi\left(\underline{x}\right)$ on H_Q^t. Again, since $\bar{\phi}_Q^t\left(\underline{x}\right)$ is affine, problem $\bar{Q}^t$ is a linear programming problem. Note that, because $\bar{Q}^t$ is a relaxation of Q^t, we have $\nu\left[\bar{Q}^t\right] \leq \nu\left[Q^t\right]$. Furthermore, if $ub\left[P\right] \geq \nu\left[Q\right]$, then for each t, we have $\nu\left[Q^t\right] = \nu\left[Q\right]$, so $\nu\left[\bar{Q}^t\right]$ is a lower bound to $\nu\left[Q\right]$. On the other hand, if $ub\left[P\right] < \nu\left[Q\right]$, then $ub\left[P\right]$ itself is a lower bound to $\nu\left[Q\right]$. This means that either $\nu\left[\bar{Q}^t\right]$ or $ub\left[P\right]$ is a lower bound to $\nu\left[Q\right]$. Therefore, for each t, we define the value CI_Q^t as

$$CI_Q^t = \min\left\{ub\left[P\right], \nu\left[\bar{Q}^t\right]\right\} \tag{4.12}$$

and the *capacity improvement lower bound* as

$$lb\,[Q] = CI_Q^t \tag{4.13}$$

The capacity improvement procedure produces a sequence of non-decreasing lower bounds, each of which is at least as tight as the relaxation lower bound (4.3). If, for any given t, the fathoming criterion (F) is satisfied using the capacity improvement lower bound (4.13), then problem Q can be fathomed. Otherwise, either action (*B1*) or action (*B2*) must be taken. If action (*B2*) is selected, the branch and bound procedure will attempt to produce a tighter lower bound to $\nu\,[Q]$ by forming and solving problem $\bar{Q}^{t+1}$, and hence computing CI_Q^{t+1}. On the other hand, if action (*B1*) is chosen, then the hyperrectangle H_Q^t will be partitioned into smaller hyperrectangles.

4.4 Calculation of Capacity Improvement Lower Bound

This Section describes a method for calculating Q^{t+1}, and hence the value of CI_Q^{t+1}, given problem Q^t. We assume that the solution to the feasibility relaxation $\bar{Q}^t$ defined in equation 4.11 is available. Since each $\bar{\phi}_{Qj}^t\,(x_j)$ is affine in $\bar{Q}^t$, the solution to $\bar{Q}^t$ is easily obtained. The solution to $\bar{Q}^t$ (in the t-th iteration) is used to determine the "improved" lower and upper bound vectors $\underline{l}_Q^{t+1}$ and $\underline{u}_Q^{t+1}$ (in the $(t+1)$-st capacity improvement iteration), thereby forming problem Q^{t+1}. Once $\underline{l}_Q^{t+1}$ and $\underline{u}_Q^{t+1}$ have been determined, problem $\bar{Q}^{t+1}$ can be solved, and $\nu\left[\bar{Q}^{t+1}\right]$ obtained. The solution to $\bar{Q}^{t+1}$ is then used to calculate CI_Q^{t+1} in equation (4.12). Thus, using this "bootstrap" method, CI_Q^t can be computed for *any* iteration number, t.

Given the solution to $\bar{Q}^t$, the improved bounds $\underline{l}_Q^{t+1}$ and $\underline{u}_Q^{t+1}$ are determined by performing a concave underestimator analysis of problem Q^t. This Section describes how Q^{t+1} and hence CI_Q^t are calculated given problem Q^t.

4.4.1 Capacity Improvement

Given the solution to problem $\bar{Q}^t$, we now wish to determine the formulation of problem $\bar{Q}^{t+1}$ used in (4.12). The formulation of problem $\bar{Q}^{t+1}$ is the same as problem $\bar{Q}^t$ except that the lower bound vectors $\underline{l}$ and $\underline{u}$ in problem $\bar{Q}^t$ are replaced with the "improved" lower and upper bound vectors $\underline{l}_Q^{t+1}$ and $\underline{u}_Q^{t+1}$ in problem $\bar{Q}^{t+1}$. The process of determining $\underline{l}_Q^{t+1}$ and $\underline{u}_Q^{t+1}$ is done element-by-element for each of these vectors. Therefore, it is sufficient to select one generic element, say the k-th element (for $k \in J$), and describe the capacity improvement procedure with respect to that particular element. Thus, for any given $k \in J$ we wish to determine lower and upper bounds, l_{Qk}^{t+1} and u_{Qk}^{t+1}, such that $l_{Qk}^{t+1} \geq l_{Qk}^t$ and $u_{Qk}^{t+1} \leq u_{Qk}^t$.

In computing l_{Qk}^{t+1} and u_{Qk}^{t+1}, we make the assumption that $ub\,[P] \geq \nu\,[Q^t]$. If this assumption is true, then the values l_{Qk}^{t+1} and u_{Qk}^{t+1} are computed such that $l_{Qk}^{t+1} \leq x_{Qk}^* \leq u_{Qk}^{t+1}$ where x_{Qk}^* is the optimal value of x_k in problem Q. In this case, $\nu\left[\hat{Q}^{t+1}\right]$ will be a lower bound to $\nu\,[Q^t]$. On the other hand, if $ub\,[P] < \nu\,[Q^t]$, then it is not necessarily true that $l_{Qk}^{t+1} \leq x_{Qk}^* \leq u_{Qk}^{t+1}$. However, in this case it is still true that the minimum of $ub\,[P]$ and $\nu\left[\hat{Q}^{t+1}\right]$ is a lower bound to $\nu\,[Q^t]$. Hence, the procedure described below is a valid method for computing CI_Q^{t+1} given in (4.12).

Recall from Chapter 3 the non-convex relaxation of Q^t, denoted $\hat{Q}^t$, as follows

$$(\hat{Q}^t) \qquad \min \hat{\phi}^t_Q(\underline{x}) \text{ s.t. } \underline{x} \in \hat{S}^t_Q = \bar{X} \cap \hat{H}^t_Q \tag{4.14}$$

where $\hat{\phi}^t_Q(\underline{x})$ was any separable objective function that satisfied properties (OB.1) to (OB.4). Recall that the parametric function $\theta^t_{Qk}(\delta_k)$ describes the change in $\nu\left[\hat{Q}^t\right]$ with the post-optimal introduction of the constraint $x^t_{Qk} < \hat{x}^t_{Qk} + \delta_k$ to problem $\hat{Q}^t$. That is,

$$\theta^t_{Qk}(\delta_k) = \nu\left[\hat{Q}^t \mid x^t_{Qk} = \hat{x}^t_{Qk} + \delta_k\right] - \nu\left[\hat{Q}^t\right]$$

The calculation of the improved lower bounds l^{t+1}_{Qk} and u^{t+1}_{Qk} for arc k is based on the function $\theta^t_{Qk}(\delta_k)$ defined in equation 3.14.

We first define two values of δ_k, denoted δ^{t+}_{Qk} and δ^{t-}_{Qk}, as follows:

$$\delta^{t+}_{Qk} = \max\left\{\delta_k : \theta^t_{Qk}(\delta_k) \leq ub[P] - \nu\left[\hat{Q}^t\right]\right\} \tag{4.15}$$

$$\delta^{t-}_{Qk} = \min\left\{\delta_k : \theta^t_{Qk}(\delta_k) \leq ub[P] - \nu\left[\hat{Q}^t\right]\right\} \tag{4.16}$$

We then have the following Theorem:

Theorem 4.4.1. *$\hat{x}^t_{Qk} + \delta^{t-}_{Qk}$ forms a lower bound to x^*_{Qk}, and $\hat{x}^t_{Qk} + \delta^{t+}_{Qk}$ an upper bound to x^*_{Qk}, where x^*_{Qk} is the value of x_k in the optimal solution to problem Q.*

Proof. We can assume that $ub\,[P] > \nu\left[\hat{Q}^t\right]$ (otherwise problem Q^t would already be fathomed using condition (F)), and, by construction, $\theta^t_{Qk}\left(\delta_k\right)$ is unimodal with $\theta^t_{Qk}\left(0\right) = 0$. This means for any $\delta_k \leq \delta^{t-}_{Qk}$ that if $ub\,[P] \geq \nu\,[Q^t]$ then $\nu\left[\hat{Q}^t \mid x^t_{Qk} < \hat{x}^t_{Qk} + \delta_k\right]$ is strictly greater than $\nu\,[Q^t]$. Moreover, since $\hat{Q}^t$ is a relaxation of Q^t, this means that, for any $\delta_k \leq \delta^{t-}_{Qk}$, $\nu\left[Q^t \mid x^t_{Qk} < \hat{x}^t_{Qk} + \delta_k\right]$ is also strictly greater than $\nu\,[Q^t]$. Thus, it must be the case that $\hat{x}^t_{Qk} + \delta^{t-}_{Qk} \leq x^*_{Qk}$. Therefore, if $ub\,[P] \geq \nu\,[Q^t]$, then $\hat{x}^t_{Qk} + \delta^{t-}_{Qk}$ forms a lower bound to x^*_{Qk}. By using similar reasoning, if $ub\,[P] \geq \nu\,[Q^t]$, it also must be the case that $\hat{x}^t_{Qk} + \delta^{t+}_{Qk} \geq x^*_{Qk}$, meaning $\hat{x}^t_{Qk} + \delta^{t+}_{Qk}$ is an upper bound to x^*_{Qk}. □

To ensure that the improved lower and upper bounds (l^{t+1}_{Qk} and u^{t+1}_{Qk}) are at least as tight as the current lower and upper bounds (l^t_{Qk} and u^t_{Qk}), we set l^{t+1}_{Qk} and u^{t+1}_{Qk} to

$$l^{t+1}_{Qk} = \max\left\{l^t_{Qk}, \hat{x}^t_{Qk} + \delta^{t-}_{Qk}\right\} \tag{4.17}$$

$$u^{t+1}_{Qk} = \min\left\{u^t_{Qk}, \hat{x}^t_{Qk} + \delta^{t+}_{Qk}\right\} \tag{4.18}$$

Equations (4.17) and (4.18) can be used for each $k \in J$ to produce the lower and upper bound vectors $\underline{l}^{t+1}_Q$ and $\underline{u}^{t+1}_Q$, and thereby forming problem $\bar{Q}^{t+1}$. Problem $\bar{Q}^{t+1}$ is solved and used to obtain $\nu\left[\bar{Q}^{t+1}\right]$. These values, in turn, are used to compute CI^{t+1}_Q in (4.12).

Recall from Chapter 3 two of the three example formulations of $\hat{\phi}^t_Q\left(\underline{x}\right)$, denoted $\hat{\phi}^L\left(\underline{x}\right)$ and $\hat{\phi}^M\left(\underline{x}\right)$. Recall also that $\hat{\phi}^L_j\left(x_j\right) \leq \hat{\phi}^M_j\left(x_j\right)$ for each

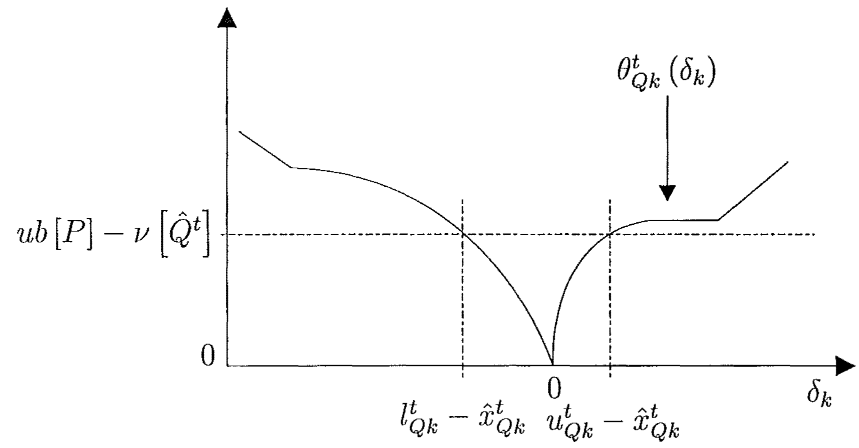

Figure 4.1: Calculation of capacity improvement parameters

$j \in J$. The first (linear) formulation, $\hat{\phi}^L(\underline{x})$, produces the standard linear capacity improvement result reported in the literature. The other (non-convex) formulation, $\hat{\phi}^M(\underline{x})$ produces improved lower bounds that are at least as tight as those produced by the linear formulation, albeit at a greater computational cost. Figure 4.1 shows the calculation of l^{t+1}_{Qk} and u^{t+1}_{Qk} for a typical arc $k \in B$ using the non-convex formulation $\hat{\phi}^M(\underline{x})$.

Finally, note that in forming the new relaxation Q^{t+1}, only the simple flow bounds on the arcs are changed. This has the important property of preserving the structure of the constraint set. For example, if the original problem P had a pure network constraint set (that is, there were no side variables or arcs and no side constraints), each subproblem Q^t would also have a pure network constraint set. Consequently, each relaxation $\bar{Q}^t$ would be a pure network flow problem.

4.4.2 Formal Algorithm Statement

The enhanced branch and bound algorithm is simply the standard branch and bound algorithm with capacity improvement performed at each branch and bound node. The enhanced branch and bound algorithm for problems of type P can be defined as follows.

Algorithm 4.4.1. Enhanced Branch and Bound with Capacity Improvement for Minimisation of problems of type P[S,LC,NBS_L,FV,RI]

Step 0.1 *Define the set* FEAS $\subset S$ *as the set of* known *feasible solutions to problem P.*

Step 0.2 *Set*

$$ub\,[P] = \begin{cases} min\,\{\phi\,(\underline{x}) : \underline{x} \in \text{FEAS}\} & \textit{if}\ \text{FEAS} \neq \emptyset \\ \infty & \textit{if}\ \text{FEAS} = \emptyset \end{cases}$$

Step 0.3 *Set* $\tilde{\underline{x}} \in$ FEAS *satisfying* $\phi\,(\underline{x}) = ub\,[P]$. *Define* $\tilde{\underline{x}}$ *as the* incumbent *solution.*

Step 0.4 *Define the set* SUB *of active subproblems to problem P as* SUB $= \{P\}$.

Step 0.5 *Set the iteration counter* $i \leftarrow 1$

Step i.1 *Select a subproblem* $Q \in$ SUB *as the current subproblem and set* SUB $=$ SUB $- \{Q\}$. *This step is known as "node selection".*

CI 0.1 *Set problem* $Q^0 = Q$

CI 0.2 *Set iteration count* $t \leftarrow 1$

CI t.1 *Form the linear program $\bar{Q}^t$ as defined in equation 4.11 and solve. Define $\underline{\overline{x}}^t$ as the solution variable vector satisfying $\bar{\phi}\{\underline{x}\} = \nu\left[\bar{Q}^t\right]$.*

CI t.2 *if $\underline{\overline{x}}^t = (\underline{\overline{x}}^t_{\mathbb{R}}, \underline{\overline{x}}^t_{\mathbb{Z}})$ such that $\underline{\overline{x}}^t_{\mathbb{R}} \in \mathbb{R}^n$ and $\underline{\overline{x}}^t_{\mathbb{Z}} \in \mathbb{Z}^m$, then update* FEAS *by setting* FEAS $\leftarrow$ FEAS $\cup\, \underline{\overline{x}}^t$.

CI t.3 *Update $ub[P] = min\{\phi(\underline{x}) : \underline{x} \in \text{FEAS}\}$ and define the incumbent solution vector $\underline{\tilde{x}}$ as the vector $\underline{x} \in$ FEAS satisfying $\phi(\underline{x}) = ub[P]$.*

CI t.4 *Delete (fathom) all subproblems $Q \in$ SUB satisfying:*

$lb[Q] \geq ub[P]$.

CI t.5 *If $\nu\left[\bar{Q}^t\right] \geq ub[P]$ subproblem Q is fathomed. Goto step* i.7.

CI t.6 *If any one of a predefined set of capacity improvement stopping criteria are met, set $Q = Q^t$ and proceed to step* i.2.

CI t.7 *Calculate the improved simple flow bounds l_j^{t+1} and u_j^{t+1} for each arc $j \in J$, and form subproblem Q^{t+1} as described in Section 4.4.*

CI t.8 *Set $t \leftarrow t+1$ and goto step* CI t.1.

Step i.2 *Construct a rectangular partition H_Q into finitely many hyper-rectangles $H_{Q_1}, \ldots, H_{Q_r}, \ldots, H_{Q_\eta}$, and form the associated subproblems of P denoted $Q_1, \ldots, Q_r, \ldots, Q_\eta$.*

Step i.3 *Obtain $lb[Q_r] = \nu\left[\bar{Q}_r\right]$, a lower bound to $\nu[Q_r]$ for each Q_r. This step is known as "bounding". Define $\underline{\overline{x}}^r = \left(\ldots, \overline{x}_j^r, \ldots\right)$ as the solution vector for problem $\bar{Q}_r$ satisfying $\phi(\underline{x}) = lb[Q_r]$.*

Step i.4 *For each* $\bar{Q}_r$*, if* $\underline{\overline{x}}^r = (\underline{\overline{x}}^r_{\mathbb{R}}, \underline{\overline{x}}^r_{\mathbb{Z}})$ *such that* $\underline{\overline{x}}^r_{\mathbb{R}} \in \mathbb{R}^n$ *and* $\underline{\overline{x}}^r_{\mathbb{Z}} \in \mathbb{Z}^m$*, update* FEAS *by setting* FEAS $\leftarrow$ FEAS $\cup\, \underline{\overline{x}}^r$*.*

Step i.5 *Update* $ub[P] = min\{\phi(\underline{x}) : \underline{x} \in \text{FEAS}\}$ *and define the incumbent solution vector* $\tilde{\underline{x}}$ *as the vector* $\underline{x} \in$ FEAS *satisfying* $\phi(\underline{x}) = ub[P]$*.*

Step i.6 *Delete (fathom) all subproblems* $Q \in$ SUB *satisfying* $lb[Q] \geq ub[P]$*.*

Step i.7 *If* SUB $\neq \emptyset$ *set the iteration counter* $i \leftarrow i+1$ *and go to step* i.1*. Otherwise,* $\nu[P] \geq ub[P]$ *and the current incumbent solution is optimal. Define* $\underline{x}^* = \tilde{\underline{x}}$ *as the optimal solution vector for problem* P *and exit the solution procedure.*

4.4.2.1 Node Selection, Bounding, and Partitioning

The node selection, bounding, and partitioning steps are as defined in Section 4.2.1.

4.4.2.2 Stopping Criteria

Any set of criteria that halts the capacity improvement algorithm after a finite number of iterations may be used. This ensures that the capacity improvement algorithm is finite. An example set of stopping criteria is to halt capacity improvement if any one of the following three conditions is met:

(i) If the proportion of basic arcs with improved bounds (that is, either $l_k^{t+1} > l_k^t$ and/or $u_k^{t+1} < u_k^t$) is less than some prespecified amount.

(ii) If the rate of increase in $\nu\left[\bar{Q}^t\right]$ as t increases is less than some prespecified rate.

(iii) If the number of capacity improvement iterations, t, is greater than a prespecified maximum number of iterations.

Condition (iii) guarantees the finiteness of the capacity improvement algorithm.

4.4.3 Convergence of the Enhanced Algorithm

The convergence of the enhanced branch and bound algorithm follows directly from the convergence of the standard branch and bound algorithm and the finiteness of the capacity improvement algorithm. We assume the set of stopping criteria used will terminate the capacity improvement procedure in a finite number of iterations. The capacity improvement procedure then adds a finite number of calculations to each node of the branch and bound tree created by the solution procedure. Further, since the capacity improvement techniques do not exclude any part of the feasible region that contains the global minimum, the convergence characteristics of the original branch and bound algorithm are preserved. Hence, if the standard branch and bound algorithm is finite, the enhanced algorithm formed by augmenting the branch and bound algorithm with the capacity improvement algorithm is also finite. Similarly, if the standard branch and bound algorithm used is convergent, the consequent enhanced branch and bound algorithm is also convergent.

Chapter 5

Computational Analysis

5.1 Introduction

In order to test the efficacy of capacity improvement, a branch and bound algorithm of the type developed in the previous Chapter was coded in C and implemented on an SGI O2 unix box. The algorithm implemented a depth–first–search node selection strategy to search the branch and bound enumeration tree. For arcs with continuous objective functions, an $\bar{x}$ subdivision strategy was used to partition the subproblems. For arcs with either a piecewise linear or fixed charge objective function, branching occurred at the piecewise linear breakpoints or at zero arc flow respectively. The LP relaxation of each subproblem was solved using the CPLEX 4.0 callable library (CPLEX Optimization Inc. 1996). Two types of capacity improvement were implemented: standard *linear* capacity improvement, calculated using the linear form of the concave underestimator analysis; and *mixed* capacity improvement, calculated using the non-convex (or mixed) form of the relaxation analysis.

Eight sets of test problems – called CONNET, TRANS, TSIDE, TINT, CLPA, CLPB, CLPC, and CMIP – were generated. The first four sets each consisted of 40 problems, and sets five through eight each consisted of 15 problems. One quarter of the test problems in the first four sets had concave quadratic objective functions, another quarter square root objective functions, a third quarter 2–piece concave piecewise linear objective functions, and the remaining quarter fixed charge objective functions. In the latter four sets, one third of the test problems had concave quadratic objective functions, another third square root objective functions, and the remaining third fixed charge objective functions. Throughout the rest of this Section, each test problem in each set is indexed by a letter indicating the objective type, and a number. The indices for the objective functions are Q (quadratic), S (square root), P (2–piece piecewise linear), and F (fixed charge). For example, CONNET–S4 refers to the fourth test problem with a square root objective function in the CONNET set. Finally, each test problem in each set with the same numerical index has the same feasible region. For example, CONNET–Q4, CONNET–S4, CONNET–P4, and CONNET–F4 differ only in the objective function – the feasible regions are identical.

In the remainder of this Section, the construction of each test problem set will be detailed and the computational results presented. Each test problem in each set was solved three times; the first with the standard branch and bound algorithm (that is, branch and bound with no capacity improvement), the second with the branch and bound algorithm using linear capacity improvement, and the third with the branch and bound algorithm using mixed capacity improvement. For each test problem set, the average CPU time in seconds, the average number of branch and bound nodes in the solution tree, and the average number of LP subproblems solved, are presented for each objective function type and capacity improvement combination. These measures provide an indication of solution speed, in-core storage requirements, and number of algorithm iterations required to obtain the optimal solution respectively. In addition, for each of the three performance measures, the average percentage improvement of the mixed capacity improvement algorithm over the linear capacity improvement algorithm for each objective function type is also listed. In all instances, the test problems were solved to optimality.

5.1.1 CONNET Test Problem Set

The first set, CONNET, was created using NETGEN, the pure network generator of Klingman, Napier & Stutz (1974). Ten pure network test problems were generated, each consisting of 25 nodes (including 2 supply nodes and 4 demand nodes) and 75 arcs. Four test problems were then created for each of the NETGEN–generated pure networks by replacing the linear objective function with a quadratic objective, a square root objective, a 2–piece piecewise linear objective, and a fixed charge objective respectively. In each case the objective function coefficients were randomly chosen from a uniform distribution with prespecified upper and lower limits as shown in Table 5.1.

Table 5.1: Cost coefficient ranges for CONNET test problem set

Function	Functional Form	Parameter Ranges
Quadratic	$c_0 + c_1\left(x - c_2\right)^2$	$c_0 = -c_1 c_2^2$ $-100 \le c_1 \le -50$ $100 \le c_2 \le 250$
Square Root	$m\sqrt{x}$	$1000 \le m \le 10000$
Fixed Charge	$\begin{cases} 0 & x = 0 \\ c_0 + c_1 x & x > 0 \end{cases}$	$10000 \le c_0 \le 50000$ $1 \le c_1 \le 25$
Piecewise Linear	$\begin{cases} c_1 x & x \le m \\ c_0 + c_2 x & x > m \end{cases}$	$-10000 \le c_1 \le 10000$ $-10000 \le c_2 \le c_1$ $m = (\text{upper bound})/2$ $c_0 > 0$

5.1.2 TRANS Test Problem Set

To construct the second set, ten transshipment test problems were generated. The networks were constructed to model a "typical" product distribution network in which goods may be shipped from several supply points to demand centres both directly or via several warehouses. Each problem consisted of 6 supply nodes, 6 warehouses, and 12 demand nodes. The supply at each supply node was randomly selected with a maximum supply of 15. Demands were randomly selected based on the total network supply, and then scaled so that demand equalled supply in the network.

Each warehouse was modelled via three nodes and three arcs, as illustrated in Figure 5.1. A fixed charge was placed on the arc from node A to node B. Warehouse capacity was modelled via an upper bound of 50 units on this arc. The first arc from node B to node C had a unit cost of zero and a set capacity that was a fraction of the upper bound on the arc from A to B. The second arc had a positive per unit (linear) variable cost. This modelled the situation, illustrated in Figure 5.2, where a fixed charge was paid for any number of goods up to a set limit, beyond which a holding and processing charge is levied per unit.

An arc was constructed from each supply node to each warehouse, from each warehouse to each demand node, and from each supply node directly to each demand node. An upper bound of 15 units was placed on each arc. Four test problems were constructed for each of the ten networks in which the transportation arc costs were concave quadratic (problems TRANS–Q1

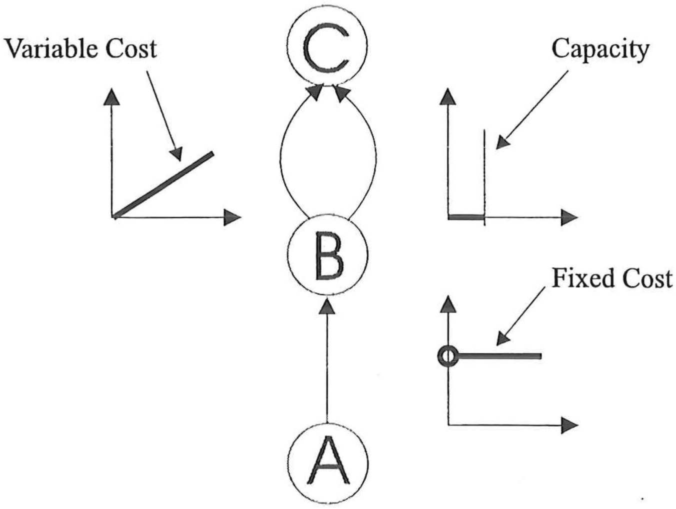

Figure 5.1: Network model of warehouse

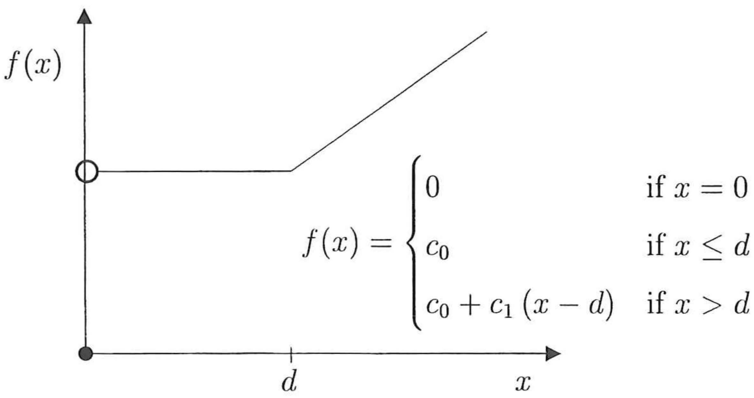

Figure 5.2: Warehouse cost function

Table 5.2: Cost coefficient ranges for TRANS test problem set

Function	Functional Form	Parameter Ranges
Quadratic	$c_0 + c_1 (x - c_2)^2$	$c_0 = -225 \times c_1$ $-2 \leq c_1 \leq -0.75$ $c_2 = 15$
Square Root	$m\sqrt{x}$	$250 \leq m \leq 600$
Fixed Charge (Transportation)	$\begin{cases} 0 & x = 0 \\ c_0 + c_1 x & x > 0 \end{cases}$	$100 \leq c_0 \leq 250$ $25 \leq c_1 \leq 50$
Fixed Charge (Warehouse)	$\begin{cases} 0 & x = 0 \\ c_0 + c_1 x & x > 0 \end{cases}$	$1000 \leq c_0 \leq 1500$ $30 \leq c_1 \leq 50$

to TRANS–Q10), square root (problems TRANS–S1 to TRANS–S10), 2–piece piecewise linear (problems TRANS–P1 to TRANS–P10), and fixed charge (problems TRANS–F1 to TRANS–F10). As in the CONNET set, the coefficients for the quadratic, square root, and fixed charge objective function coefficients were randomly chosen from a uniform distribution with prespecified upper and lower limits as given in Table 5.2. The objective function on each arc in the piecewise linear set of TRANS test problems (TRANS–P1 to TRANS–P10) was a 2–piece piecewise linear approximation of the quadratic cost function on that arc in the corresponding problem in the TRANS-Q test problem set.

5.1.3 TSIDE Test Problem Set

The set TSIDE was formed simply by adding a set of side constraints to each test problem from the second test problem set, TRANS. The side constraints were used to ensure that the throughput of each warehouse was within 20% of the throughput of the "base" (taken to be the first) warehouse. Two side constraints were added for each warehouse other than the first to enforced this condition. In all other respects, the test problems in the two sets TRANS and TSIDE were identical.

5.1.4 TINT Test Problem Set

The set TINT was created by taking each test problem from the TRANS set and enforcing the condition that each supply node may supply the demand nodes either directly or via the warehouses but not both. This was achieved via a binary variable for each supply node that had a value of either 0 or 1 depending on whether the goods were shipped directly from that supply node to the demand nodes or the goods were shipped via the system of warehouses. A side constraint was added for each supply node to enforce the either/or condition.

5.1.5 CLPA Test Problem Set

As was the case in the construction of the CONNET test problem set, The fifth set, CLP, was formed using the NETGEN network test problem generator. Five pure network test problems were generated, each with two supply nodes, four demand nodes, eleven transshipment nodes, and fifty arcs. Ten randomly generated side constraints were then added to each of the networks. Three test problems were then created for each of the five problems by replacing the original linear objective function on each arc with

Table 5.3: Cost coefficient ranges for CLPA test problem set

Function	Functional Form	Parameter Ranges
Quadratic	$c_0 + c_1 (x - c_2)^2$	$c_0 = 0$ $-80 \leq c_1 \leq -70$ $225 \leq c_2 \leq 275$
Square Root	$m\sqrt{x}$	$900 \leq m \leq 1100$
Fixed Charge	$\begin{cases} 0 & x = 0 \\ c_0 + c_1 x & x > 0 \end{cases}$	$24000 \leq c_0 \leq 26000$ $20 \leq c_1 \leq 30$

a quadratic, fixed charge, and square root objective function respectively. Once again, the objective function coefficients were randomly chosen from a uniform distribution with upper and lower limits given in Table 5.3.

5.1.6 CLPB Test Problem Set

The set CLPB was formed in exactly the same manner as CLPA, except only five side constraints per problem were generated. Thus each problem in set CLPB had the same objective function parameters as the corresponding problem in set CLPA, but had a somewhat less constrained feasible region.[1]

5.1.7 CLPC Test Problem Set

The seventh set, CLPC, consisted of the test problems from set CLPA with the objective functions replaced by objective functions with coefficients generated from a uniform distribution with parameters given in Table 5.4. Note that the objective functions for each test problem in set CLPC are much

[1]That is, a feasible region with relatively more structure.

Table 5.4: Cost coefficient ranges for CLPC test problem set

Function	Functional Form	Parameter Ranges
Quadratic	$c_0 + c_1 (x - c_2)^2$	$c_0 = 0$ $c_1 = -2$ $450 \leq c_2 \leq 550$
Square Root	$m\sqrt{x}$	$m = 4$
Fixed Charge	$\begin{cases} 0 & x = 0 \\ c_0 + c_1 x & x > 0 \end{cases}$	$900 \leq c_0 \leq 1100$ $20 \leq c_1 \leq 30$

"flatter" (that is, the area between each separable objective and its lower convex envelope is less) than the objective functions in the corresponding problem from set CLPA.

5.1.8 CMIP Test Problem Set

The CMIP test problem set was created using the same methodology as the CLP test problem set. Five pure network problems were generated with the NETGEN generator, each with two supply nodes, four demand nodes, eleven transshipment nodes, and thirty arcs. Five randomly generated side constraints were then added to each of the networks, and every second arc was defined to be integer valued. Three test problems were then created for each of the five problems by replacing the original linear objective function on each arc with a quadratic, fixed charge, and square root objective function respectively. The objective function coefficients were chosen from the same distributions as those used in the generation of the CLPA set (Table 5.3).

5.2 Computational Results and Analysis

5.2.1 CONNET Test Problem Set

From the computational results in Tables 5.5, 5.6 and 5.7 two facts are immediately apparent. The first is that, when compared to the standard branch and bound algorithm with no capacity improvement, implementing any form of capacity improvement offers substantial improvement in solution time and requirements. The second is that mixed capacity improvement offers a significant performance increase, both in terms of solution time and in–core storage requirements, over that provided by linear capacity improvement.

However, the actual advantage offered by mixed capacity improvement was dependent upon the functional form of the objective function. Specifically, mixed capacity improvement provided a reasonably modest improvement in solution time for the 2-piece piecewise linear and quadratic test problems of approximately 7%. More significant gains were observed for the fixed charge and square root test problems (17% and 34% respectively). Similar results were observed for size of the branch and bound tree and the number of linear programs solved performance measures.

5.2.2 TRANS, TSIDE, and TINT Test Problem Sets

Again, as for the CONNET test problem set, the results from the computational testing on the TRANS, TSIDE, and TINT test problem sets indicates that a vast performance increase over the standard branch and bound algorithm can be expected with either form of capacity improvement. Again,

Table 5.5: Average CPU time (seconds) for CONNET problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 3600	482.74	426.84	93.65%
Fixed Charge	16.91	8.29	6.67	84.62%
Square Root	24.79	3.41	2.65	74.50 %
Piecewise Linear	41.81	21.98	20.51	93.41%

Table 5.6: Average number of branch and bound nodes for CONNET problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 500000	42958.2	38270.2	93.84%
Fixed Charge	6521.3	748.4	600.8	78.38%
Square Root	10425.8	172.6	146.4	81.97%
Piecewise Linear	16519.8	1838.6	1773.2	94.67%

Table 5.7: Average number of LPS for CONNET problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 500000	141955.1	124225.1	91.98%
Fixed Charge	6521.3	2375.1	1745.0	78.57%
Square Root	10425.8	954.7	708.2	71.43%
Piecewise Linear	16519.8	6187.0	5549.4	90.50%

Table 5.8: Average CPU time (seconds) for TRANS problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	102.83	5.43	5.48	101.57%
Fixed Charge	> 3600	73.01	72.65	96.82%
Square Root	> 3600	24.50	23.80	95.53%
Piecewise Linear	> 2321	7.22	7.63	101.48%

it is also apparent that for each test problem set, mixed capacity improvement provided similar and modest gains for those problems with 2-piece piecewise linear and quadratic objectives, whilst providing more substantial performance increases for the fixed charge and square root problems.

Further, comparing the results for each objective function type across the three test problem groups, it is apparent that increasing the complexity of the feasible region, whether it be via the introduction of side constraints or integer side variables and associated side constraints into the network formulation, increases the performance of mixed relative to linear capacity improvement. For example, for the test problems with square root objectives, mixed capacity improvement provided a solution time increase over linear capacity improvement of approximately 5% for the TRANS test problems, 9% for the TSIDE problems, and 14% for the TINT problems.

5.2.3 CLPA, CLPB, and CLPC Test Problem Sets

First, note that for the three CLP test problem sets, in all but a few cases the standard branch and bound solution algorithm failed to find and confirm the optimal solution within 3,600 CPU seconds. As in the previous

Table 5.9: Average number of branch and bound nodes for TRANS problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	26370.6	237	242.8	104.02%
Fixed Charge	> 500000	4047.0	4005.8	97.31 %
Square Root	> 500000	1317.4	1209.8	89.70%
Piecewise Linear	> 490000	309.2	319.8	100.02%

Table 5.10: Average number of LPS for TRANS problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	26370.6	592.2	586.1	99.78%
Fixed Charge	> 500000	8368.6	8099.1	94.38%
Square Root	> 500000	2683.4	2464.0	90.44%
Piecewise Linear	> 490000	742.0	773	99.32%

Table 5.11: Average CPU time (seconds) for TSIDE problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 3600	24.01	21.15	95.46%
Fixed Charge	> 3600	266.26	224.99	84.76%
Square Root	> 3600	90.76	81.75	91.64%
Piecewise Linear	> 3323	16.12	15.43	99.41%

Table 5.12: Average number of branch and bound nodes for TSIDE problem set

Problem Set	Capacity Improvement			Mixed as % of Linear
	None	Linear	Mixed	
Quadratic	> 500000	525.8	450.0	91.16%
Fixed Charge	> 500000	9068.6	7585.6	84.19%
Square Root	> 500000	3312.8	2725.4	85.44%
Piecewise Linear	> 500000	459.2	414.8	94.91%

Table 5.13: Average number of LPS for TSIDE problem set

Problem Set	Capacity Improvement			Mixed as % of Linear
	None	Linear	Mixed	
Quadratic	> 500000	1878.7	1611.6	92.55%
Fixed Charge	> 500000	20850.5	17267.2	83.39%
Square Root	> 500000	7363.3	6247.9	87.16%
Piecewise Linear	> 500000	1229.7	1134.2	96.73%

Table 5.14: Average CPU time (seconds) for TINT problem set

Problem Set	Capacity Improvement			Mixed as % of Linear
	None	Linear	Mixed	
Quadratic	265.28	25.12	24.59	97.21%
Fixed Charge	> 3600	215.75	205.62	95.09%
Square Root	> 3600	119.75	105.96	87.98%
Piecewise Linear	> 3021	33.58	34.94	96.69%

Table 5.15: Average number of branch and bound nodes for TINT problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	30607.7	602.8	604.6	102.30%
Fixed Charge	> 500000	6744.4	6362.2	93.65%
Square Root	> 500000	3054.8	2642.6	85.70%
Piecewise Linear	> 500000	945.0	981.6	96.27%

Table 5.16: Average number of LPS for TINT problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	30607.7	1299.9	1262.4	96.67%
Fixed Charge	> 500000	13928.5	12949.6	92.69%
Square Root	> 500000	7671.0	6745.0	87.51%
Piecewise Linear	> 500000	1951.2	1980.3	94.59%

results, it is observed that mixed capacity improvement offers a larger performance increase over linear capacity improvement for the square root and fixed charge test problems than it does for those problems with quadratic objectives. However, this difference is less pronounced than for the previous test problem sets.

Examining the raw results for each objective function type across the series of sets yields the unsurprising result that decreasing the complexity of the feasible region (from CLPA to CLPB) decreases the solution time and storage requirements. Similarly, "flattening" the objective functions (CLPA to CLPC) also decreased the solution time and storage requirements.

The performance of mixed capacity improvement relative to linear capacity improvement was variable across both the test problem sets and the types of objective functions. For the quadratic and square root test problems, mixed capacity improvement provided the best performance increase for set CLPA, followed by CLPC, followed by CLPB. However, for the fixed charge problems, the relative performance of mixed capacity improvement was highest for the CLPB set, followed by the CLPA set, (closely) followed by the CLPC set. However, for all three objective function types, mixed capacity improvement performed relatively better on the CLPA set than the CLPC set by a small margin. From this we may draw the tentative conclusion that the structure of the feasible region is of greater import than the "flatness" or otherwise of the objective function in determining the performance of mixed relative to linear capacity improvement.

Table 5.17: Average CPU time (seconds) for CLPA problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 3600	60.51	54.24	89.57%
Fixed Charge	> 3600	160.23	137.72	87.98%
Square Root	> 3600	281.41	229.24	80.57%

Table 5.18: Average number of branch and bound nodes for CLPA problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 500000	3535.4	3141.0	86.16%
Fixed Charge	> 500000	11127.8	9536.2	87.84%
Square Root	> 500000	12649.4	10574.6	82.13%

Table 5.19: Average number of LPS for CLPA problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 500000	15747.6	14137.8	89.67%
Fixed Charge	> 500000	46418.4	39413.0	87.15%
Square Root	> 500000	73777.8	58780.4	82.13%

Table 5.20: Average CPU time (seconds) for CLPB problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 3600	45.12	43.50	96.32%
Fixed Charge	> 3600	48.96	39.00	79.82%
Square Root	> 3600	138.72	121.69	88.79%

Table 5.21: Average number of branch and bound nodes for CLPB problem set

Problem Set	Capacity Improvement			Mixed as % of Linear
	None	Linear	Mixed	
Quadratic	> 500000	3712.2	3321.8	89.80%
Fixed Charge	> 500000	3851.4	3154.2	81.89%
Square Root	> 500000	7765.8	6759.0	85.79%

Table 5.22: Average number of LPS for CLPB problem set

Problem Set	Capacity Improvement			Mixed as % of Linear
	None	Linear	Mixed	
Quadratic	> 500000	12978.2	12194.8	93.66%
Fixed Charge	> 500000	14548.8	11554.4	80.08%
Square Root	> 500000	38832.2	33254.0	87.60%

Table 5.23: Average CPU time (seconds) for CLPC problem set

Problem Set	Capacity Improvement			Mixed as % of Linear
	None	Linear	Mixed	
Quadratic	> 3600	28.16	24.45	92.95%
Fixed Charge	> 3600	9.00	7.95	90.76%
Square Root	> 3600	140.29	112.77	80.45%

Table 5.24: Average number of branch and bound nodes for CLPC problem set

Problem Set	Capacity Improvement			Mixed as % of Linear
	None	Linear	Mixed	
Quadratic	> 500000	1245.8	1077.0	89.91%
Fixed Charge	> 500000	388.2	343.0	90.13%
Square Root	> 500000	4906.6	4080.8	81.45%

Table 5.25: Average number of LPS for CLPC problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 500000	7965.4	6754.0	91.27%
Fixed Charge	> 500000	2695.4	2350.8	90.00%
Square Root	> 500000	40420.2	31234.2	77.42%

5.2.4 CMIP Test Problem Set

In the final test problem set, CMIP, the pattern of results are somewhat different. Specifically, we can observe that mixed relative to linear capacity improvement performed better on the quadratic test problems than both the fixed charge and square root test problems. This was not due to a degradation in performance over problems with square root or fixed charge objectives; rather, mixed capacity improvement out-performed linear capacity improvement by a much larger margin than in the previous test problem sets (an average percentage solution time improvement of 28% rather than 0% to 14%). One possible reason for this apparent anomaly is that the fixed charge (and to a lesser extent, the square root) problems were solved on average much faster than the quadratic problems. Consequently, mixed capacity improvement was given more opportunity to out-perform linear capacity improvement. However, this pattern has not been noticed in the previous test problem sets; the relative performance of mixed over linear capacity improvement did not appear to be greatly related to the raw solution time.

Table 5.26: Average CPU time (seconds) for CMIP problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 3600	60.79	42.82	77.92%
Fixed Charge	> 3600	7.01	6.15	87.65%
Square Root	> 3600	23.46	19.32	87.21%

Table 5.27: Average number of branch and bound nodes for CMIP problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 500000	1873.8	1493.0	87.27%
Fixed Charge	> 500000	560.2	496.6	88.86%
Square Root	> 500000	1045.6	943.4	91.96%

Table 5.28: Average number of LPS for CMIP problem set

Problem Set	Capacity Improvement			Mixed as
	None	Linear	Mixed	% of Linear
Quadratic	> 500000	10372.0	8232.6	83.65%
Fixed Charge	> 500000	2233.0	1980.4	87.80%
Square Root	> 500000	6848.4	5680.4	87.09%

5.3 Conclusions

In conclusion, from the computational analysis it is apparent that either form of capacity improvement provides a significant performance increase over the case where no capacity improvement is used, and that mixed capacity improvement provides a more modest, but still significant, performance increase over linear capacity improvement. The level of this performance increase appears to be dependent upon two aspects of the problem in particular: the complexity of the objective function, and the complexity of the feasible region.

Concentrating on the effect of the objective function, in general the results from the computational analysis suggest two things. First, as the difference between the objective function and its convex envelope increased for each particular objective function type, the relative performance of mixed capacity improvement also increased. Because mixed capacity improvement uses the actual objective function in its analysis, its comparative advantage over linear capacity improvement, which uses only the convex envelope cost information, increases. Second, mixed capacity improvement performed better relative to linear capacity improvement for fixed charge and square root objective functions than it did for quadratic and 2-piece piecewise linear objective functions. This is possibly due to the fact that the "slope"[2] of the fixed charge and square root objectives changes rapidly for small changes in arc flow when the arc flow is small. For quadratic and 2-piece piecewise

[2]In a subgradient sense.

linear objectives, the change in slope is much more gradual. Again, because mixed capacity improvement uses the actual objective function in its analysis, it may have an advantage over linear capacity improvement.

Similarly, as the feasible region of the problem becomes more "complex" or constrained the performance of mixed relative to linear capacity improvement appears to improve. This can be seen by comparing the results from the TRANS set with those from the TSIDE and TINT sets, and the results from the CLPB set with those from the CLPA set. In addition, for problems in the CLP series of test problem sets, it appears that the structure of the feasible region is of greater import than the "flatness" of the objective function (for a given objective function type) in determining the performance of mixed relative to linear capacity improvement.

In summary, we can make the following conclusions:

(I) Any form of capacity improvement provides a significant performance increase over the case where no capacity improvement is used.

(II) Mixed capacity improvement provides, on average, a more modest but still significant performance increase over linear capacity improvement.

(III) The more complex the feasible region, the greater the performance increase offered by mixed over linear capacity improvement.

(IV) The greater the difference between the objective function and its lower convex envelope, the greater the performance increase offered by mixed over linear capacity improvement.

(V) The greater the rate of change of "slope" of objective function, the greater the performance increase offered by mixed over linear capacity improvement.

(VI) For a given problem, the structure of the feasible region appears more important than the "flatness" of the objective function in determining the performance of mixed capacity improvement relative to linear capacity improvement.

Chapter 6

Power Dispatch with Piecewise Linear Losses

6.1 Introduction

Over the past decade, the New Zealand electricity industry has undertaken a process of reform and deregulation, with the aim of establishing a fully competitive wholesale electricity market structure. A similar process has recently been undertaken in the Australian and several segments of the U.S. power industries.

One important aspect of such markets is the short–term dispatch of generation and load connected to the transmission network, and the consequent calculation of prices for electricity at all points (called *nodes* or *buses*)[1] in the network[2]. In both the New Zealand and the Australian markets,

[1]Throughout this Chapter, the terms "nodes" and "buses" will be used interchangeably.

[2]The price at a node is a quantity that reflects the marginal cost of supplying electricity to that node. As such, it implicitly incorporates the cost of generation and the cost

such a dispatch is determined in the following general manner: "Blocks" or "amounts" of generation, load, and reserve[3], each with an associated bid price, are offered into the market by producers and consumers connected to the transmission grid. The market is then "cleared" by solving an optimisation model of the dispatch process, and the corresponding nodal prices are determined. Based on the result of the optimisation, the market players may revise their offers, and the process is repeated.

The optimisation model used to determine the dispatch is a linear program that maximises the net benefit of generation subject to the system constraints. Net benefit is defined as the revenue obtained from supplying electricity less the cost of supply. The cost of supply consists of the cost of generation, plus the cost of providing reserve. The constraints include standard DC power flow equations, and constraints governing spinning reserve, system risk, and system security[4]. Transmission losses are modelled via a set of constraints that determine the transmission loss as a piecewise linear function of power flow. Nodal prices are given by the value of the

of transmission losses and all constraints associated with generation and transmission. For a complete exposition of nodal pricing of electricity, see Read & Ring (1995).

[3]Reserve is the term given to generation capacity used to provide "cover" for a system failure, such as the failure of a generator or transmission line. Reserve can be of three types: partially loaded spinning reserve (*PLSR*), tail-water depressed (*TWD*) reserve, and interruptible load (*IR*). Partially loaded spinning reserve can be defined as the extra output level that a generator can attain within a specified time frame (usually within several seconds) following a sudden drop in the AC frequency of the transmission system (the system frequency). A turbine run in tail-water depressed mode spins at the system frequency by drawing a small amount of power, but uses no water. Obviously, only hydro generators can provide TWD reserve. Machines running in TWD mode can be used to aid the recovery of the system frequency. Interruptible load is load that can be quickly "shed" from the system, and can only be provided by purchasers.

[4]The Grid Operator may impose generation and/or flow limits on transmission equipment for security reasons.

dual variables corresponding to the nodal power flow balance constraints governing the power flow into and out of the network nodes.

Normally, the LP model correctly provides a dispatch that is optimal to the LP and is physically implementable. However, under certain conditions, the LP no longer correctly models the physical situation, and hence the "optimal" dispatch given by the LP may not correspond to an implementable, or physical, dispatch. In this Section, we first provide a mathematical description of the LP dispatch model. For reasons of confidentiality the model will be described in general terms only; however, this does not effect the subsequent analysis. Following this, the conditions under which a non-physical dispatch may arise will be detailed. A solution methodology for this problem that models the dispatch problem as an MIP and uses mixed capacity improvement as part of the solution algorithm is then proposed. Finally, a small numerical example demonstrating the solution approach is presented.

6.2 Linear Programming Model

Underlying the LP dispatch model is a linear load flow model that represents the flow of electricity in the transmission network. Power flow in an electrical transmission network must obey both Kirchhoff's current law and voltage law, defined as follows:

Definition 6.2.1. Kirchhoff's Current Law

The sum of the currents entering a point in an electric circuit must equal the sum of the currents leaving that point.

Definition 6.2.2. Kirchhoff's Voltage Law

In any closed electric circuit, the sum of the voltage drops across the circuit elements must equal the sum of the applied electromotive force around every closed loop.

Kirchhoff's current law is modelled with the nodal power flow balance constraints[5]

$$\underline{g} - \underline{d} - A^T \cdot \underline{f} = \underline{0} \tag{6.1}$$

where $\underline{g} = \{\ldots, g_i, \ldots\}$ is the generation vector with g_i denoting the total generation at node i, $\underline{d} = \{\ldots, d_i, \ldots\}$ is the demand vector with d_i denoting the total demand for electricity at node i, and $\underline{f} = \{\ldots, f_k, \ldots\}$ is the vector of transmission line flows, and A is the node–arc incidence matrix for the transmission network with generic element $a_{k,i}$. Element $a_{k,i} = 1 = -a_{k,j}$ if line k directly connects bus (node) i and bus (node) j, and 0 otherwise[6]. The flow on each transmission line has physical flow limits defined by

$$-u_k \leq f_k \leq u_k \tag{6.2}$$

where u_k is the maximum permissible flow on line k.

Kirchhoff's voltage law is modelled as

$$\underline{f} - \Omega \cdot A \cdot \underline{\delta} = \underline{0} \tag{6.3}$$

[5]Note that the nodal power flow balance constraints correspond to the conservation of flow constraints in a pure network flow model.

[6]Therefore $f_k < 0$ implies flow from node j to node i on line k, and $f_k > 0$ implies flow from node i to node j on line k.

where Ω is the diagonal matrix with diagonal element Ω_k being the impedance of line k, and $\underline{\delta} = \{\ldots, \delta_i, \ldots\}$ where δ_i denotes the voltage angle variable at node i relative to the swing bus[7], where by definition for the swing bus $\delta_{swing} = 0$.

Transmission losses on a line are modelled as a piecewise linear function of line flow. That is, the loss on line k, denoted l_k is given by

$$l_k = \underline{t}_k \cdot \underline{f}_k + l_{\underline{k}} \tag{6.4}$$

where $l_{\underline{k}}$ is the loss when $f_k = -u_k$ (that is, the loss at the largest possible transfer from node j to i), $s = (1, \ldots, \bar{s}_k)$ indexes the loss tranches[8] on line k, $\underline{t}_k = \{\ldots, t_k^s, \ldots\}$ is the vector of loss coefficients for line k with t_k^s denoting the coefficient associated with the s-th loss tranche for line k and $t_k^s < t_k^{s+1}$. Typically $t_k^s < 0$ if tranche s corresponds to a negative value of f_k, and $t_k^s > 0$ if tranche s corresponds to a positive value of f_k. The vector $\underline{f}_k = \{\ldots, f_k^s, \ldots\}$ is the vector of tranche flows on line k where f_k^s is the flow on line k associated with the s-th loss tranche and

$$f_k = \sum_s f_k^s - u_k \tag{6.5}$$

and

[7]Power flow on a transmission line is proportional to the difference in voltage angle at the sending bus (or node) and receiving bus (or node). That is, it is the differences in the values of the nodal voltage angle variables that are important in the DC load flow model, rather than the absolute values. Hence, voltage angles are measured with respect to the voltage angle at a reference bus, called the swing bus.

[8]The s-th loss tranche on line k is that segment of the piecewise linear transmission loss function for which the marginal loss is t_k^s.

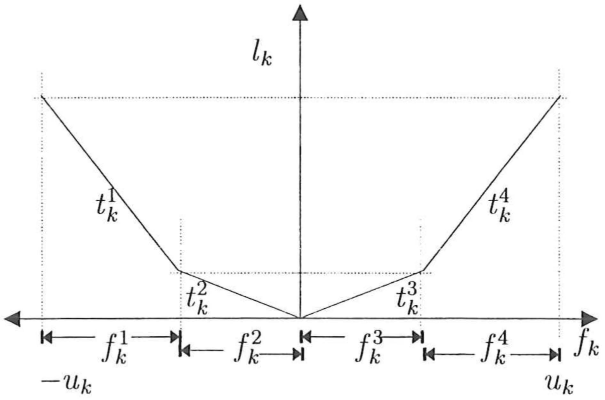

Figure 6.1: Example piecewise linear transmission loss function

$$0 \leq f_k^s \leq u_k^s \ \forall \ s \tag{6.6}$$

where u_k^s is the maximum power flow on tranche s of line k, and

$$\sum_s u_k^s = 2 \times u_k \tag{6.7}$$

Figure 6.1 describes an example of a loss function on a transmission line.

Typically, the transmission losses are added to the linear load flow model by incorporating losses into the conservation of power flow constraints (equation (6.1)) as follows:

$$\underline{g} - \underline{d} - A \cdot \underline{f} - 0.5|A| \cdot \underline{l} = \underline{0} \tag{6.8}$$

where $|A|$ is the matrix with generic element $|a_{k,i}|$; that is, the element in the i-th row and j-th column of matrix $|A|$ is the absolute value of the element in the i-th row and j-th column of matrix A[9]. Finally, the risk, security, and reserve constraints are represented via

$$C\underline{z} = \underline{b} \tag{6.9}$$

where $\underline{z}$ is the vector representing all the decision variables in the model.

The objective is to maximise the net benefit from generation. That is

$$\max \quad \underline{\phi}(\underline{z}) = \underline{\delta}(\underline{d}) - \underline{\gamma}(\underline{g}) - \underline{\rho}(\underline{r}) \tag{6.10}$$

where $\underline{\delta}(\underline{d}) = \sum_j \delta_j(d_j)$ is a separable concave piecewise linear function giving the revenue from supplying electricity, $\underline{\gamma}(\underline{g}) = \sum_j \gamma_j(d_j)$ is a separable convex piecewise linear function giving the cost of generation, and $\underline{\rho}(\underline{r}) = \sum_j \rho_j(d_j)$ is a separable convex piecewise linear function giving the cost of reserve. The objective function $\underline{\phi}(\underline{z})$ is thus separable concave piecewise linear. The concave piecewise linearity of the objective function results from the market rules which specify generation, reserve and demand offers to be in the form of a set of bands, or "bid blocks", each with an associated constant marginal price.

Thus, for a given set of generation, load, and reserve bids, the dispatch model discussed in this Section is a linear program, and can be solved to produce a dispatch schedule. The dual variables from the LP–optimal

[9] That is, half the loss on line k is allocated to node i, and half to node j. Alternatively, the losses on a line may be allocated completely to the sending or receiving node. Thus, if $f_k > 0$, l_k would be allocated to node i, and if $f_k < 0$, l_k would be allocated to node j. This would require a slightly different formulation of the losses involving either two loss or two flow variables for each line – one for when $f_k > 0$ and the other for when $f_k < 0$.

schedule (that is, the schedule obtained by solving the linear programming dispatch model) can be used to provide pricing information; for example, the dual variable associated with a conservation of power flow constraint gives the price of electricity at the corresponding node.

6.3 Non–Physical Dispatch

A non–physical dispatch is a dispatch schedule that cannot be achieved in reality. Such a schedule can arise in the LP dispatch model described above because the implied loss cost function is not convex. The problem arises when the price of energy at one or more nodes becomes negative.[10]

Two common causes of the non–physical loss problem are generation offers at negative prices, and constrained loops in cyclic transmission networks. The former arises in predominantly large fossil fuel or nuclear generation systems wherein it may be expensive for a generator to generate below a certain level. In such a situation, the generator may offer power at a negative price; that is, the generator is prepared to pay to be able to remain "on" and generate at (or above) some minimum level. If loads fall so low that all generators must operate below their minimum generation level, and if such a generator becomes marginal [11] energy prices may become

[10] That is, the optimal values of the dual variable(s) associated with one (or more) nodal flow balance constraint is negative if the objective is a maximisation, or positive if the objective is a minimisation.

[11] The marginal generator is the generator that would supply a marginal increase in generation required in the network. For example, assuming no losses and no transmission constraints, the generation would be dispatched in order of price (merit order), and the marginal generator would be the generator that is partially loaded (i.e. generating at a level less that its upper bound).

negative and the LP will implicitly maximise generation. For a given load, using high loss tranches before low loss tranches increases the apparent level of generation in the LP model; however such a solution clearly would not correspond to a physical dispatch and therefore cannot be implemented.

A cyclic, or meshed, transmission network is one in which transmission lines form loops or cycles. A phenomenon called the *spring washer effect*, in which energy prices may become negative at some point in a loop, can arise in such networks. The spring washer effect is described fully in Read & Ring (1995); the following discussion is a summary of their analysis. Consider first a simple cyclic network in which a loop has power entering at one node and exiting at another. From Kirchhoff's (Voltage) Law, current entering the loop will split itself between the two paths; the proportion "taking" each path is directly related to the line impedances on each of the two paths. If the loop is unconstrained[12] the relative prices in the loop depend only upon transmission losses, and will rise along all paths from the entering node to the exiting node. Any increase in demand will be met at the marginal node in the loop[13].

Assume now that the demand for power at the more expensive "exiting" side of the loop is increasing. This extra demand will be supplied by the marginal node, with the power flow dividing between the two paths as dictated by the line impedances. However, if one of the transmission lines in the loop becomes constrained[14], it will be impossible for the one marginal

[12]That is, no transmission line in the loop is at its capacity.

[13]There could be a partially loaded generator at this node, or a transmission line connecting the node to a partially loaded generator outside the loop.

[14]That is, the power flow on the line is at the line's capacity.

generator alone to provide the extra power without increasing the flow on the constrained line. Hence a second marginal generator will be required. If this new marginal node is connected between the old marginal node and the upstream side of the constraint[15], the new marginal node will have to reduce its net injection into the loop, thereby relieving pressure on the upstream side of the constraint. Alternatively, if the new marginal node is connected between the old marginal node and the downstream side of the constraint, the new marginal node will have to increase its net injection into the loop, thereby increasing power flow to the downstream side of the constraint, and thus reducing flow across it. In either situation, the price will rise on the downstream side of the constraint, and decrease on the upstream side[16] where it may even become negative[17]. In such a situation, there will be an incentive for the LP to use loss tranches on affected lines out of order.

A simple example illustrating the effect of negative energy prices is as follows. Consider the simple network consisting of a single line connecting two buses (nodes) denoted by the indices i and j, along which 10 MW of power must flow from i to j. Losses are modelled by a two–piece piecewise linear formulation, where the first loss tranche has marginal losses of 2% and a capacity of 6 MW, and the second tranche a marginal loss of 5% and a capacity of 8 MW. Figure 6.2 illustrates such a network. Assuming the

[15]We describe flow across the constraint as being from the "upstream" side to the "downstream" side.

[16]It may be convenient to consider a dispatcher decreasing the price to encourage demand and discourage generation on the upstream side, and increasing the price to encourage generation and discourage demand on the downstream side.

[17]This reflects the situation in which the dispatcher would be prepared to pay for increased load as the best way to relieve pressure on a transmission line.

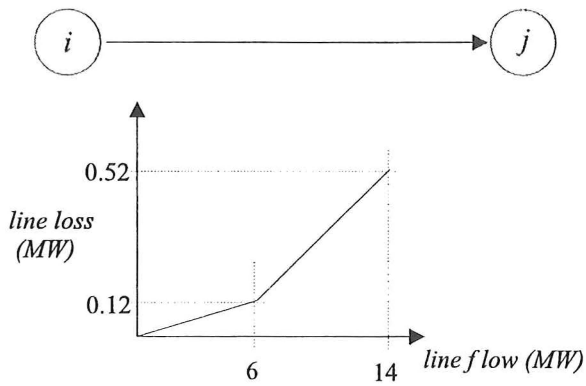

Figure 6.2: Example transmission line with piecewise linear losses

price of energy on the transmission line is \$1/MW[18], the LP will minimise losses by assigning 6MW of flow to the first tranche, and 4 MW of flow to the second tranche, with total losses of 0.32MW. This is also the correct physical solution. However, setting the price of energy to -\$1/MW gives an incentive to the LP to set losses as high as possible for a given line flow. In this situation, the LP will assign 8 MW of flow to the second tranche, and 2 MW of flow to the first, giving a total flow of 10 MW and a total loss of 0.44 MW. In this case, the losses from the LP solution do not correspond to the physical losses on the line. In a model of a realistic transmission system, the non–physical loss problem can give rise to a dispatch schedule in which large quantities of power effectively vanish from the system.

[18]That is, this is the marginal price of energy implied by the dual of the dispatch model. For simplicity, no differentiation is made between losses at i and losses at j.

6.4 Solution Method for the Dispatch Problem

The dispatch problem may be formulated as follows:

$$
\begin{aligned}
(Q) \quad & \min \quad -\underline{\phi}\left(\underline{z}\right) && (6.11)\\
& \text{s.t.} \quad H\underline{z} = \underline{h} && (6.12)\\
& \qquad p_k\left(f_k\right) = l_k \;\; \forall\, k && (6.13)\\
& && (6.14)
\end{aligned}
$$

where $H\underline{z} = \underline{h}$ is the set of all constraints (including lower and upper bound definitions) except those defining line loss as a piecewise linear function of line flow, $\underline{\phi}\left(\underline{z}\right)$ is the objective function of the LP dispatch model, and $p_k(f_k)$ is the piecewise linear loss function for line k defined as

$$p_k\left(f_k\right) = \underline{l_k} + \sum_{s \leq \bar{s}_k} \left(t_k^s \cdot u_k^s\right) + t_k^{\bar{s}+1} \cdot \left(f_k - \sum_{s \leq \bar{s}_k} u_k^s\right) \qquad (6.15)$$

where $\bar{s}_k$ is the tranche index defined as

$$\bar{s}_k = \max\left\{s : \sum_s u_k^s \leq f_k\right\} \qquad (6.16)$$

and u_k^s is the maximum flow on tranche s of line k. Note that the feasible region of problem Q implied by equation (6.13) is nonconvex.

6.4.1 Mixed–Integer Programming Model

One approach to the solution of the dispatch problem is to formulate the dispatch problem as a mixed–integer program. For each flow tranche s_k on each transmission line k we define a binary variable denoted β_k^s and recall that variable f_k^s gives the flow on tranche s_k on line k. The dispatch model can then be formulated as the following mixed–integer program:

$$
\begin{array}{llll}
(Q_{\mathrm{MIP}}) & \min & -\underline{\phi}\,(\underline{z}) & \\
 & \text{s.t.} & H\underline{z} \quad = \underline{h} & \\
 & & l_{\underline{k}} + \underline{t}_k \cdot \underline{f}_k = l_k & \forall\, k \\
 & & \sum_s f_k^s - u_k = f_k & \forall\, k \\
 & & f_k^s \quad \geq \beta_k^s \cdot u_k^s & \forall\, k, s \\
 & & f_k^s \quad \leq \beta_k^{s-1} \cdot u_k^s & \forall\, k, s \\
 & & \beta_k^s \quad \in \{0, 1\} & \forall\, k, s
\end{array}
$$

Problem Q_{MIP} can then be solved using any standard MIP solution algorithm. Moreover, structure inherent in the dispatch problem may be exploited to increase the performance of the MIP solution algorithm. In the following Section, the theory of capacity improvement is extended to take advantage of the piecewise linear structure of the loss constraints in the dispatch model.

6.4.2 Capacity Improvement in the Dispatch Model

In the following, we assume that we have a subproblem of the dispatch problem MIP Q_{MIP}, denoted by $\tilde{Q}_{\mathrm{MIP}}$.

The linear programming relaxation of the dispatch model MIP $\tilde{Q}_{\mathrm{MIP}}$ is simply formed by relaxing the integrality requirement on the (0,1) variables. We denote this relaxation as problem $\bar{Q}_{\mathrm{MIP}}$ defined as

$$
\begin{array}{llll}
(\bar{Q}_{\mathrm{MIP}}) & \min & -\underline{\phi}\,(\underline{z}) & \\
& \text{s.t.} & H\underline{z} = \underline{h} & \\
& & l_{\underline{k}} + \underline{t}_k \cdot \underline{f}_k = l_k & \forall\, k \\
& & \sum_s f_k^s - u_k = f_k & \forall\, k \\
& & f_k^s \geq \beta_k^s \cdot u_k^s & \forall\, k, s \\
& & f_k^s \leq \beta_k^{s-1} \cdot u_k^s & \forall\, k, s \\
& & \beta_k^s \geq 0 & \forall\, k, s \\
& & \beta_k^s \leq 1 & \forall\, k, s
\end{array}
$$

From Lagrangian duality theory, problem $\bar{Q}_{\mathrm{MIP}}$ is related to the linear program, denoted $\bar{P}_{\underline{\pi}}$, formed by augmenting the objective function of problem $\bar{Q}_{\mathrm{MIP}}$ with each loss constraint multiplied by the value of its corresponding dual variable in the optimal solution to $\bar{Q}_{\mathrm{MIP}}$. That is, $\bar{P}_{\underline{\pi}}$ is the linear program defined as

$$
\begin{aligned}
(\bar{P}_{\underline{\bar{\pi}}}) \qquad & \min \quad -\underline{\phi}(\underline{z}) + \sum_k \bar{\pi}_k \cdot \left(l_k - l_{\underline{k}} - \underline{t}_k \cdot \underline{f}_k\right) \\
& \text{s.t.} \quad H\underline{z} && = \underline{h} \\
& \sum_s f_k^s - u_k = f_k && \forall\, k \\
& f_k^s && \geq \beta_k^s \cdot u_k^s \quad \forall\, k, s \\
& f_k^s && \leq \beta_k^{s-1} \cdot u_k^s \quad \forall\, k, s \\
& \beta_k^s && \geq 0 \quad \forall\, k, s \\
& \beta_k^s && \leq 1 \quad \forall\, k, s
\end{aligned}
$$

where $\underline{\pi} = \{\ldots, \pi_k, \ldots\}$ is the dual variable associated with the loss constraint on line k, and $\bar{\pi}_k$ is the value of π_k in the optimal solution to the linear programme $\bar{Q}_{\mathrm{MIP}}$. Next, we form another problem, denoted $P_{\underline{\bar{\pi}}}$, based on problem $\bar{P}_{\underline{\bar{\pi}}}$ as follows:

$$
\begin{aligned}
(P_{\underline{\bar{\pi}}}) \qquad & \min \quad -\underline{\phi}(\underline{z}) + \sum_k \bar{\pi}_k \cdot (l_k - p_k(f_k)) \\
& \text{s.t.} \quad H\underline{z} = \underline{h}
\end{aligned}
$$

It is immediately observable that problem $P_{\underline{\bar{\pi}}}$ has a nonlinear separable objective function term for each flow variable f_k when $\bar{\pi}_k \neq 0$. In fact, for those variables f_k where $\bar{\pi}_k$ is positive, the function $-\bar{\pi}_k \cdot p_k(f_k)$ is concave piecewise linear. Similarly, for those variables f_k where $\bar{\pi}_k$ is negative, the function $-\bar{\pi}_k \cdot p_k(f_k)$ is convex piecewise linear. This, combined with the fact that a minimum cost convex piecewise linear objective function can be modelled as a series of linear functions, implies problem $P_{\underline{\pi}}$ can be viewed as a separable concave cost minimisation problem. Further, note that for those variables f_k where $\bar{\pi}_k > 0$, the objective function $-\bar{\pi}_k \cdot \left(l_{\underline{k}} + \underline{t}_k \cdot \underline{f}_k\right)$ combined with the constraints from problem $\bar{P}_{\underline{\pi}}$ describes the lower convex envelope of the objective function $-\bar{\pi}_k \cdot p_k(f_k)$, and hence problem $\bar{P}_{\underline{\pi}}$ is a linear relaxation of problem $P_{\underline{\pi}}$.

In summary, we have a separable concave cost minimisation problem (problem $P_{\underline{\pi}}$) and a linear relaxation of $P_{\underline{\pi}}$ created by replacing each concave objective function with its lower convex (affine) envelope (problem $\bar{P}_{\underline{\pi}}$). Using the capacity improvement techniques of Chapter 4, we can obtain new upper and lower bounds for each f_k, denoted b_k^u and b_k^l respectively, for which $\nu\left[P_{\underline{\pi}}|f_k \leq b_k^l\right] \geq INC_Q$ and $\nu\left[P_{\underline{\pi}}|f_k \geq b_k^u\right] \geq INC_Q$ where problem "$P_{\underline{\pi}}|\cdot$" denotes problem $P_{\underline{\pi}}$ augmented with the constraint "$\cdot$", and INC_Q denotes the objective function value of the incumbent solution to the dispatch model Q_{MIP}.

We now consider the problem "$P_{\underline{\pi}}|f_k \leq b_k^l$", and form a new problem, denoted "$P_{\pi}|f_k \leq b_k^l$", that is identical to problem "$P_{\underline{\pi}}|f_k \leq b_k^l$" except that the price vector $\underline{\pi}$ is allowed to vary. Clearly then,

$$\nu\left[P_{\underline{\pi}}|f_k \leq b_k^l\right] \leq \max_{\underline{\pi}}\left\{\nu\left[P_{\underline{\pi}}|f_k \leq b_k^l\right]\right\} \tag{6.17}$$

and thus

$$\max_{\underline{\pi}}\left\{\nu\left[P_{\underline{\pi}}|f_k \leq b_k^l\right]\right\} \geq INC_Q \tag{6.18}$$

Problem "$P_{\underline{\pi}}|f_k \leq b_k^l$" is a partial dual to problem "$\tilde{Q}_{MIP}|f_k \leq b_k^l$", the dispatch problem $\tilde{Q}_{MIP}$ augmented with the constraint $f_k \leq b_k^l$. Therefore

$$\nu\left[\tilde{Q}_{MIP}|f_k \leq b_k^l\right] \geq \max_{\underline{\pi}}\left\{\nu\left[P_{\underline{\pi}}|f_k \leq b_k^l\right]\right\} \geq INC_Q \tag{6.19}$$

That is, there is no solution to problem $\tilde{Q}_{\text{MIP}}$, in which $f_k \leq b_k^l$, that has a smaller objective function value than the current incumbent solution. Thus, b_k^l is a valid lower bound for f_k in the current subproblem $\tilde{Q}_{\text{MIP}}$ of the dispatch model Q_{MIP}. Similarly, it can also be shown that b_k^u is a valid upper bound for f_k in the current subproblem $\tilde{Q}_{\text{MIP}}$.

6.4.3 Numerical Example

We consider the network formed by three buses, denoted by the letters B, C and D, and a transmission line connecting buses B and D, denoted line BD, and a transmission line connecting buses C and D, denoted line CD. Bus D has a load of 400 MW. A generator at bus B offers three 100 MW blocks of generation, the first at a price of -\$5/MW, the second -\$4/MW, and the third at -\$2/MW. A generator at bus C offers four 100 MW blocks of generation, the first at a price of -\$3/MW, the second at -\$1/MW, the third at \$1/MW, and the fourth at \$4/MW. Line BD has a capacity of 250 MW, with a marginal loss of 6 % for the first 100 MW, 13 % for power flow from 100 MW to 200 MW, and 22 % for flows above 200 MW. Line CD has a capacity of 350 MW, with a marginal loss of 4 % for the first 100 MW, 8 % for power flow from 100 MW to 200 MW, 13 % for flow from 200 MW to 300 MW, and 20 % for flows above 300 MW. For simplicity,

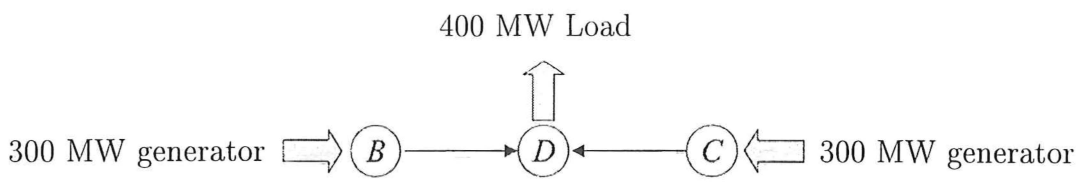

Figure 6.3: Example transmission network

losses are considered to be incurred at the sending bus only. The objective is to minimise the cost of supplying power to the load at bus D. Figure 6.3 illustrates such a network.

The dispatch model for this network can then be formulated as follows. First, for each of the two generation buses we construct a conservation of power flow constraint as follows:

$$\sum_k g_i^k - f_{iD} = 0 \tag{6.20}$$

where decision variable g_i^k is the level of generation from the k^{th} block of generation at bus i, and f_{iD} gives the power flow on the transmission line from bus i to bus D as measured at D. As the direction of power flow on each line is known, losses can be incorporated in the conservation of flow constraint as follows:

$$\sum_k g_i^k - f_{iD} - p_{iD}(f_{iD}) = 0 \tag{6.21}$$

where $p_{iD}(f_{iD})$ is a piecewise linear function giving the transmission loss on line iD as a function of the power flow on line iD. For line BD, we have

$$p_{BD}(f_{BD}) = \begin{cases} 0.06 \times f_{BD} & \text{if } f_{BD} \leq 100 \\ 6 + 0.13 \times (f_{BD} - 100) & \text{if } 100 < f_{BD} \leq 200 \\ 19 + 0.22 \times (f_{BD} - 200) & \text{if } 200 < f_{BD} \leq 250 \end{cases} \tag{6.22}$$

and for line CD we have

$$p_{CD}(f_{CD}) = \begin{cases} 0.04 \times f_{CD} & \text{if } f_{CD} \leq 100 \\ 4 + 0.08 \times (f_{CD} - 100) & \text{if } 100 < f_{CD} \leq 200 \\ 12 + 0.13 \times (f_{CD} - 200) & \text{if } 200 < f_{CD} \leq 300 \\ 25 + 0.20 \times (f_{CD} - 300) & \text{if } 300 < f_{CD} \leq 350 \end{cases} \tag{6.23}$$

The conservation of flow constraint for the load bus, bus D, is simply

$$f_{BD} + f_{CD} = 400 \tag{6.24}$$

With the addition of simple variable bounds and a minimum cost objective function, the dispatch problem for this simple network, denoted problem Q, is given by

$$
\begin{array}{llll}
(Q) & \min\ z_{LP} = & -5g_B^1 - 4g_B^2 - 2g_B^3 - 3g_C^1 - 1g_C^2 + 1g_C^3 + 4g_C^4 & \\
& \text{s.t.} & \sum_{k=1}^{3} g_B^k - f_{BD} - p_{BD}(f_{BD}) = 0 & \\
& & \sum_{k=1}^{3} g_C^k - f_{CD} - p_{CD}(f_{CD}) = 0 & \\
& & f_{BD} + f_{CD} = 400 & \\
& & g_B^k \leq 100 & \forall\, k = 1,2,3 \\
& & g_C^k \leq 100 & \forall\, k = 1,2,3,4 \\
& & f_{BD} \leq 250 & \\
& & f_{CD} \leq 350 &
\end{array}
$$

Note that, as there are no closed loops in this example network, Kirchhoff's second law can be ignored in this model, with the flows being represented directly in terms of megawatts (MW).

The linear programming relaxation of the dispatch model can easily be formulated by replacing the piecewise linear loss functions with its concave (affine) envelope[19]. That is, the linear programming relaxation of the dispatch model Q, denoted problem $\bar{Q}$, is given by:

[19] That is, an affine loss function that intersects the piecewise linear loss function at the upper and lower line flow bounds, and overestimates the loss at any line flow between these bounds. Due to the simple structure of this example, it can be easily seen that the energy prices will be negative, and hence the functions formed by multiplying the piecewise linear loss functions by the appropriate energy price will be concave. Hence it is valid to effectively replace each loss function by its concave (affine) envelope to form the LP relaxation. However, in general the full relaxed MIP formulation used in the previous Section is required.

Table 6.1: Optimum solution to LP relaxation of example transmission network

Variable	Value
z_{LP}	-1425
g_B^1	100
g_B^2	100
g_B^3	80
g_C^1	100
g_C^2	65
f_{BD}	250
f_{CD}	150

Table 6.2: Optimum simplex tableau for LP relaxation of example transmission network

g_B^1	g_B^2	g_B^3	g_C^1	g_C^2	g_C^3	g_C^4	f_{BD}	f_{CD}	RHS	Dual
-1	-1	1	0	0	0	0	1.12	0	80	2.0
0	0	0	-1	1	1	1	-1.1	0	65	1.0
0	0	0	0	0	0	0	-1	1	150.00	1.1
3	2	0	2	0	2	5	1.14	0	-1425	–

$$
\begin{array}{lll}
(\bar{Q}) & \min\ z_{LP} = & -\ 5g_B^1 - 4g_B^2 - 2g_B^3 - 3g_C^1 - 1g_C^2 + 1g_C^3 + 4g_C^3 \\
\text{s.t.} & & \sum_{k=1}^{3} g_B^k - 1.12 f_{BD} = 0 \\
& & \sum_{k=1}^{3} g_C^k - 1.1 f_{CD} \ \ = 0 \\
& & f_{BD} + f_{CD} \qquad\quad = 400 \\
& & g_B^k \qquad\qquad\qquad \leq 100 \quad \forall\, k = 1,2,3 \\
& & g_C^k \qquad\qquad\qquad \leq 100 \ \forall\, k = 1,2,3,4 \\
& & f_{BD} \qquad\qquad\quad\ \ \leq 250 \\
& & f_{CD} \qquad\qquad\quad\ \ \leq 350
\end{array}
$$

The optimal solution to problem $\bar{Q}$ is given in Table 6.1. Table 6.2 gives the optimal simplex tableau for problem LP. Clearly, the optimum LP objective value of -1475 gives a lower bound to the optimal objective value of the dispatch problem. Obtaining an upper bound can be a more difficult process, however. One option can be to take the dispatch from the previous period, adjusting for (presumably small) changes in load and available generation. Alternatively, it may be possible obtain a feasible dispatch by simply "correcting" the linear programming solution by calculating the "true" losses for the LP–optimal power flows, and adjusting the generation levels to compensate. In this example, we assume that in the previous period there was a load of 400 MW at bus D, with 188 MW of flow transmitted on line CD (with $g_B^1 = g_B^2 = 100$ MW), and the remainder transmitted on line BD. Such a solution is a feasible dispatch in the current period, and has an objective value of -1379.4872.

We now form a Lagrangian relaxation, denoted problem $\bar{P}_{\bar{\pi}}$, of the network by augmenting the objective function of problem $\bar{Q}$ with the conservation of flow constraints for buses B and C multiplied by the value in the optimal solution to problem $\bar{Q}$ of their respective dual variables. That is,

$$
\begin{aligned}
(\bar{P}_{\bar{\pi}}) \qquad \min\ z_{LP} = & - 5g_B^1 - 4g_B^2 - 2g_B^3 - 3g_C^1 - 1g_C^2 + 1g_C^3 \\
& - 2 \times \left(1.14 f_{BD} - g_B^1 - g_B^2 - g_B^3\right) \\
& - 1 \times \left(1.1 f_{BD} - g_C^1 - g_C^2 - g_C^3 - g_C^4\right) \\
= & - 3g_B^1 - 2g_B^2 - 2g_C^1 + 2g_C^3 + 5g_C^4 - 2.24 f_{BD} - 1.1 f_{CD} \\
\text{s.t.} \qquad & f_{BD} + f_{CD} = 400 \\
& g_B^k \qquad \le 100 \ \forall\, k = 1,2,3 \\
& g_C^k \qquad \le 100 \ \forall\, k = 1,2,3 \\
& f_{BD} \qquad \le 250 \\
& f_{CD} \qquad \le 350
\end{aligned}
$$

Problem $\bar{P}_{\bar{\pi}}$ is a linear programming relaxation of the following concave minimisation problem:

$$
\begin{aligned}
(P_{\bar{\pi}}) \quad \min\ z_{LP} = & -3g_B^1 - 2g_B^2 - 2g_C^1 + 2g_C^2 + 5g_C^4 \\
& -2(f_{BD} + p_{BD}(f_{BD})) - (f_{CD} + p_{CD}(f_{CD})) \\
\text{s.t.} \quad & \sum_{k=1}^{3} g_C^k - 1.1 f_{CD} = 0 \\
& f_{BD} + f_{CD} \qquad = 400 \\
& g_B^k \qquad \leq 100 \ \ \forall\, k = 1,2,3 \\
& g_C^k \qquad \leq 100 \ \ \forall\, k = 1,2,3 \\
& f_{BD} \qquad \leq 250 \\
& f_{CD} \qquad \leq 350
\end{aligned}
$$

New lower and upper bounds for variables f_{BD} and f_{CD} can be obtained by performing mixed capacity improvement on problem $P_{\bar{\pi}}$. In the optimal solution to problem $\bar{P}_{\bar{\pi}}$, variable f_{BD} is non-basic at its upper bound. The parametric function $\theta_{BD}^M(\delta_{BD})$ is defined as

$$\theta_{BD}^{M}(\delta_{BD}) = \begin{cases} \Delta_{BD}^{M}(u_{BD} - \delta_{BD}) - \Delta_{BD}^{M}(u_{BD}) & \delta_{BD} \leq 0 \\ +\infty & \delta_{BD} > 0 \end{cases} \tag{6.25}$$

$$= \begin{cases} \hat{\underline{\phi}}_{BD}^{M}(u_{BD} - \delta_{BD}) - \hat{\underline{\phi}}_{BD}^{M}(u_{BD}) & \delta_{BD} \leq 0 \\ \quad - \bar{z}_{BD}\delta_{BD} & \\ +\infty & \delta_{BD} < 0 \end{cases} \tag{6.26}$$

$$= \begin{cases} -1.34 \times \delta_{BD} & -50 \leq \delta_{BD} \leq 0 \\ 67 - 1.16 \times (\delta_{BD} + 50) & -150 < \delta_{BD} \leq -50 \\ 183 - 1.02 \times (\delta_{BD} + 150) & -250 < \delta_{BD} \leq -150 \\ -1.14 \times \delta_{BD} & \delta_{BD} < -250 \\ +\infty & \delta_{BD} > 0 \end{cases} \tag{6.27}$$

Recall from Chapter 3 that $\theta_{BD}^{M}(\delta_{BD})$ gives an underestimate of the increase in the optimal objective function value above $\nu\left[P_{\bar{\pi}}\right]$ for problem $P_{\bar{\pi}}$ augmented with the constraint $f_{BD} = \bar{f}_{BD} + \delta_{BD}$, where $\bar{f}_{BD}$ is the value of f_{BD} in the optimal solution to problem $\bar{P}_{\bar{\pi}}$. Recall from Chapter 4, $\theta_{BD}^{M}(\delta_{BD})$ can be used to calculate new mixed capacity improvement lower and upper bounds for f_{BD}, denoted l_{BD}^{new} and u_{BD}^{new} respectively, as

$$\begin{aligned} l_{BD}^{new} &= \bar{f}_{BD} + \min\left\{\delta : \theta_{BD}\left(\delta\right) \leq INC_{Q} - \nu\left[\bar{Q}\right]\right\} \\ &= 250 - 39.7436 \div 1.34 \\ &= 220.3406 \end{aligned} \tag{6.28}$$

and

$$\begin{aligned} u_{BD}^{new} &= \bar{f}_{BD} + \max\left\{\delta : \theta_{BD}(\delta) \leq INC_Q - \nu\left[\bar{Q}\right]\right\} \\ &= 250 + 39.7436 \div \infty \\ &= 250 \end{aligned} \tag{6.29}$$

That is, in the optimal solution to the example dispatch problem the value of f_{BD} is $220.3406 \leq f_{BD} \leq 250$, compared to $0 \leq f_{BD} \leq 250$ in the original formulation. Similarly, for the basic variable f_{CD} we have:

$$\theta_{CD}^{M}(\delta_{CD}) = \begin{cases} 1.34 \times \delta_{CD} & 0 \leq \delta_{CD} \leq 50 \\ 67 + 1.16 \times (\delta_{CD} - 50) & 50 < \delta_{CD} \leq 150 \\ 183 + 1.02 \times (\delta_{CD} - 150) & 150 < \delta_{CD} \leq 250 \\ 1.14 \times \delta_{CD} & \delta_{CD} > 250 \\ +\infty & \delta_{CD} < 0 \end{cases} \tag{6.30}$$

and therefore

$$\begin{aligned} l_{CD}^{new} &= \bar{f}_{CD} + \min\left\{\delta : \theta_{CD}(\delta) \leq INC_Q - \nu\left[\bar{Q}\right]\right\} \\ &= 150 - 39.7436 \div \infty \\ &= 150 \end{aligned} \tag{6.31}$$

and

$$\begin{aligned} u_{CD}^{new} &= \bar{f}_{CD} + \max\left\{\delta : \theta_{CD}(\delta) \leq INC_Q - \nu\left[\bar{Q}\right]\right\} \\ &= 150 + 39.7436 \div 1.34 \\ &= 179.6594 \end{aligned} \tag{6.32}$$

That is, in the optimal solution to the example dispatch problem the value of f_{CD} is $150 \leq f_{CD} \leq 179.6594$, compared to $0 \leq f_{CD} \leq 350$ in the original formulation. The ranges for both f_{BD} and f_{CD} now lie entirely on one loss tranche, and hence the solution to the linear program

$$
\begin{array}{llll}
(LP) & \min\ z_{LP} = & -\,5g_B^1 - 4g_B^2 - 2g_B^3 - 3g_C^1 - 1g_C^2 + 1g_C^3 + 4g_C^4 & \\
& \text{s.t.} & \sum_{k=1}^{3} g_B^k - 1.12 f_{BD} = 0 & \\
& & \sum_{k=1}^{3} g_C^k - 1.08 f_{CD} = 0 & \\
& & f_{BD} + f_{CD} \qquad = 400 & \\
& & g_B^k \qquad \leq 100 & \forall\, k = 1,2,3 \\
& & g_C^k \qquad \leq 100 & \forall\, k = 1,2,3,4 \\
& & f_{BD} \qquad \leq 250 & \\
& & f_{BD} \qquad \geq 220.3406 & \\
& & f_{CD} \qquad \leq 179.6594 & \\
& & f_{CD} \qquad \geq 150 &
\end{array}
$$

will provide the optimal primal solution to the dispatch problem in the current period.

Chapter 7

Summary and Future Research

7.1 Summary

This thesis has been concerned with the analysis, solution, and application of nonconvex network flow problems. To this end, this thesis began by introducing nonconvex network flow models in Chapter 1. Specifically, the class of problems considered in this thesis, minimum cost network flow problems with separable nonconvex objective functions, linear and non-linear side constraints, and continuous and integer variables, was formulated.

Chapter 2 gave a brief review of the literature on solution methods for the class of problems considered in this thesis. Branch and bound algorithms, enumerative procedures such as extreme point ranking and cutting plane techniques, outer approximation, and convexification algorithms were discussed. Real–world applications of nonconvex network flow models reported in the operations research literature were reviewed. Examples of such applications included modelling of investment in and operation of waste–disposal and management systems, gas pipeline system design, and the design of communication networks.

The concept of concave underestimators was introduced in Chapter 3. A theoretical framework based on concave underestimators for the analysis of nonconvex network flow problems of the type P(S,LC,NB$S_L S_N$,FV,RI) was then developed. A specific relaxation utilising the concave underestimator of the original concave objective function was presented. This problem had two important properties: first, its optimal solution was exactly the optimal solution to a linear programming relaxation of the original problem that was (relatively) easy to solve; and second, post-optimal parametric analysis was readily performed.

Chapter 4 of the thesis developed an application of concave underestimator analysis called enhanced capacity improvement. Enhanced capacity improvement can be used as part of an algorithm to solve mixed-integer concave cost network flow problems with side constraints. A branch and bound algorithm incorporating enhanced capacity improvement was developed.

The results of the computational testing of the algorithm were presented in Chapter 5. Based on this analysis, it was concluded that:

(I) Any form of capacity improvement provided a significant performance increase over the case where no capacity improvement is used.

(II) Mixed capacity improvement provided, on average, a more modest but still significant performance increase over linear capacity improvement.

(III) The more complex the feasible region, the greater the performance increase offered by mixed over linear capacity improvement.

(IV) The greater the difference between the objective function and its convex envelope, the greater the performance increase offered by mixed over linear capacity improvement.

(V) The greater the rate of change of "slope" of objective function, the greater the performance increase offered by mixed over linear capacity improvement.

(VI) For a given problem, the structure of the feasible region appears more important than the "flatness" of the objective function in determining the performance of mixed capacity improvement relative to linear capacity improvement.

Chapter 6 formulated and developed an algorithm for the solution of the short term electricity transmission dispatch problem. A DC ("direct current") power flow formulation was used as the basis of the model formulation. Transmission losses were incorporated in the model via a piecewise linear function of transmission line flow. A solution methodology for this problem that modelled the dispatch problem as an MIP and used mixed capacity improvement as part of the solution algorithm was developed. A small numerical example was presented to demonstrate the solution approach.

7.2 Future Research

Four lines of future research are readily apparent. The first is the extension of concave underestimator analysis to more general objective functions that the separable concave functions considered in Chapter 3. The second is extending concave underestimator analysis to explicitly take account of non-convex feasible regions. The third is constructing underestimator formulations other than the linear, concave, and mixed formulations presented in Chapter 3. Finally, the fourth avenue is developing solution techniques in addition to capacity improvement using concave underestimator analysis.

From a theoretical perspective, the first of these research avenues would appear to be the most important. The linear form of the concave underestimator analysis can be extended to the class of nonconvex network flow problems with non–separable objective functions in a relatively easy manner. It is well known in the literature that a *linear* programming relaxation of a nonconvex minimisation problem can be easily constructed over a simplex containing the feasible region of the original problem (see, for example, Horst et al. (1995)). This linear programming relaxation can then be analysed using the methods of Chapter 3. However, such a straight forward extension of concave underestimator analysis in general is not readily apparent. As the class of problems with non-separable objective functions is large, the extension of concave underestimator analysis to these type of problems is an important direction of future research.

The second avenue of research, extending concave underestimator analysis to account for non-convexity in the feasible region, is suggested by the analysis of Chapter 6. Using such techniques to recognise the relationship between integer programming, and minimisation problems with non-convex objective functions and/or non-convex feasible regions can be beneficial in the development of algorithms for both classes of problems. For example, the analysis of Chapter 6 could be used in an MIP code to take advantage of any non-convex piecewise linearity in the constraint set.

The third research avenue, constructing alternative underestimator formulations other than those presented in Chapter 3, would appear to be less promising. Bell, Lamar & Wallace (1997) develop a form of capacity improvement for fixed charge problems that is of the form of the "mixed" formulation presented in Chapter 3 with the addition of a capacity constraint on the non–basic arcs. This new relaxation can be considerably tighter than the mixed formulation; however it is also more computationally intensive to calculate. Computational testing of the capacity improvement algorithm based on this new formulation in Bell et al. (1997), whilst indicating that the new formulation has a definite, albeit small, computational advantage over the mixed formulation, indicates that further advances in this area will be subject to the "law" of diminishing returns. That is, very quickly the computation effort required to calculate more complex underestimator formulations will outweigh any advantage these formulations give.

The fourth avenue of research, that of developing other applications of concave underestimator analysis, is important from an algorithmic perspective. Additional techniques based on concave underestimator analysis have the potential to provide benefits in terms of solution speed and/or in-core storage requirements to algorithms for solving nonconvex network flow problems of the class considered in this thesis. As an illustration of this avenue of research, the application of concave underestimator analysis to penalties is considered in the following Section.

7.2.1 Application of Concave Underestimator Analysis to Penalties

Recall the branch and bound solution algorithms presented in Chapter 4 for problems of type P[S,LC,NBS_L,FV,RI]. In the algorithm, a particular subproblem, Q, can be removed from consideration, or fathomed, if the following criteria holds:

$$(F) \qquad lb\,[Q] \geq ub\,[P] \tag{7.1}$$

where

$$lb\,[Q] = \nu\left[\bar{Q}\right] \tag{7.2}$$

and $\bar{Q}$ is the LP relaxation of subproblem Q. An alternative fathoming condition, called the penalty fathoming criterion, is given by:

$$(PF) \qquad lb\,[Q] + PEN_Q \geq ub\,[P] \tag{7.3}$$

where PEN_Q, the penalty associated with subproblem Q, is a non–negative scalar. A penalty PEN_Q is called a *valid penalty* (Bretthauer 1994) if fathoming criterion (PF) holds only when problem Q does not contain a solution better than the current incumbent solution. Penalties can thus be used to provide a tighter fathoming criterion than that provided by (F). The size of the branch and bound tree created by the solution procedure, and the time required to obtain the optimal solution, can thereby be reduced.

Three types of penalties that occur in the literature are conditional penalties, cut penalties, and combined conditional and cut penalties. Conditional penalties provide a lower bound to $\nu[Q]$ by obtaining an underestimate of the objective function value increase that arises from executing the partitioning branching action (*B1*) (see Section 4.2). Examples of such penalties are the "up-and-down" penalties for integer programming proposed by Driebeek (Driebeek 1966) and extended by Cabot & Erenguc (1986), Tomlin (1971), Palekar, Karwan & Zionts (1990) Lamar & Wallace (1997), and Bell et al. (1997).

Cut penalties give a lower bound on the optimal objective function value increase that would occur with the addition of a cutting plane, such as that proposed in Tuy (1964), to the subproblem. Penalties based on cutting planes were introduced in Bretthauer et al. (1994) and Bretthauer (1994). The former develops a general theory of cut penalties based upon a linear relaxation of an original (non–separable) concave minimisation problem. The theory is then applied to form a penalty, based on the Tuy cutting plane, for concave integer minimisation. The second paper presents the Tuy penalty for continuous concave minimisation problems.

The third type of penalty considers the effects of conditional and cut penalties in tandem; that is, it provides a lower bound by obtaining an underestimate of the objective function value increase that arises from executing branching action (*B1*) when a cutting plane has also been added to the subproblem. An example of this type of penalty is presented in Bell & Lamar (1995).

All three penalty types have in common the post-optimal parametric analysis of a linear relaxation of the original nonconvex optimisation problem. The penalties may therefore be extended via the post-optimal parametric analysis of a concave underestimator based relaxation of the original problem. The resulting penalties would be larger (or "tighter") than those obtained using the traditional linear relaxation approach. Implemented as part of a branch and bound algorithm they have the potential to greatly reduce the size of the branch and bound tree, thereby decreasing solution time. Such "modified" penalties are therefore worthy of future research.

Bibliography

Al-Khayyal, F. A. & Larsen, C. (1990), 'Global optimization of a quadratic function subject to a bounded mixed integer constraint set', *Annals of Operations Research* **25**, 169–180.

Bell, G. J. & Lamar, B. W. (1995), A new penalty for concave minimisation over a polytope, *in* 'Proceedings of the 31^{st} Annual Conference of the Operational Research Society of New Zealand', pp. 127–134.

Bell, G. J., Lamar, B. W. & Wallace, C. A. (1997), Capacity improvement, penalties, and the fixed charge transportation problem. Submitted to Naval Research Logistics.

Benson, H. P. (1982), 'On the convergence of two branch–and–bound algorithms for nonconvex programming problems', *Journal of Optimization Theory and Applications* **36**, 129–134.

Benson, H. P. (1990), 'Separable concave minimization via partial outer approximation and branch and bound', *Operation Research Letters* **9**, 389–394.

Benson, H. P. (1995), Concave minimization: Theory, applications and algorithms, *in* R. Horst & P. M. Pardalos, eds, 'Handbook of Global

Optimization', Klewer Academic Publishers, Dordrecht, pp. 43–148.

Benson, H. P. (1996), 'Deterministic algorithms for constrained concave minimization: A unified critical survey', *Naval Research Logistics* **43**, 765–795.

Benson, H. P. & Erenguc, S. S. (1990), 'An algorithm for concave integer minimisation over a polyhedron', *Naval Research Logistics* **37**, 515–525.

Benson, H. P., Erenguc, S. S. & Horst, R. (1990), 'A note on adapting methods for continuous global optimization to the discrete case', *Annals of Operations Research* **25**, 243–252.

Benson, H. P. & Horst, R. (1991), 'A branch and bound–outer approximation algorithm for concave minimisation over a concave set', *Journal of Computers and Mathematics with Applications* **21**, 67–76.

Benson, H. P. & Sayin, S. (1994), 'A finite concave minimisation algorithm using branch and bound and neighbor generation', *Journal of Global Optimization* **5**, 1–14.

Bertsekas, D. P. (1979), 'Convexification procedures and decomposition methods for nonconvex optimization problems', *Journal of Optimization Theory and Applications* **29**, 169–197.

Bloemhof-Ruwaard, J. M., Salomon, M. & van Wassenhove, L. N. (1996), 'The capacitated distribution and waste disposal problem', *European Journal of Operational Research* **88**, 490–503.

Blumenfeld, D. E., Burns, L. D., Daganzo, C. F., Frick, M. C. & Hall, R. W. (1987), 'Reducing logistics costs at general motors', *Interfaces* **17**, 26–47.

Bretthauer, K. M. (1994), 'A penalty for concave minimisation derived from the tuy cutting plane', *Naval Research Logistics* **41**, 455–463.

Bretthauer, K. M. & Cabot, A. V. (1994), 'A composite branch and bound, cutting plane algorithm for concave minimisation over a polytope', *Computers and Operations Research* **21**, 777–785.

Bretthauer, K. M., Cabot, A. V. & Venkataramanan, M. A. (1994), 'An algorithm and new penalties for concave integer minimisation over a polyhedron', *Naval Research Logistics* **41**, 435–454.

Cabot, A. V. (1974), 'Variations on a cutting plane method for solving concave minimisation problems with linear constraints', *Naval Research Logistics Quarterly* **21**, 265–274.

Cabot, A. V. & Erenguc, S. S. (1986), 'A branch and bound algorithm for solving a class of nonlinear integer programming problems', *Naval Research Logistics Quaterly* **33**, 559–567.

Caruso, C., Colorni, A. & Paruccini, M. (1993), 'The regional solid waste management system: A modelling approach', *European Journal of Operational Research* **70**, 16–30.

Charnes, A. & Cooper, W. W. (1961), *Management Models and Industrial Applications of Linear Programming*, Wiley, New York.

CPLEX Optimization Inc. (1996), *Using the Cplex Callable Library*, CPLEX Optimization Inc., Incline Village, New York.

Daellenbach, H. G., George, J. A. & McNickle, D. C. (1983), *Introduction to Operations Research Techniques*, second edn, Allyn and Bacon, Inc., Newton, Massachusetts.

Driebeek, N. (1966), 'An algorithm for the solution of mixed integer programming problems', *Management Science* **12**, 576–587.

Falk, J. E. & Hoffman, K. R. (1976), 'A successive underestimation method for concave minimization problems', *Mathematics of Operations Research* **1**, 251–259.

Falk, J. E. & Soland, R. M. (1969), 'An algorithm for separable nonconvex programming problems', *Management Science* **15**, 550–569.

Fetterolf, P. C. & Anandalingam, G. (1992), 'A lagrangion relaxation technique for optimizing interconnection of local area networks', *Operations Research* **40**, 678–688.

Florian, M. & Robillard, P. (1971), 'An implicit enumeration algorithm for the concave cost network flow problem', *Management Science* **18**, 184–193.

Geoffrion, A. M. & Marsten, R. E. (1972), 'Integer programming alogrithms: A framework and state–of–the–art survey', *Management Science* **18**, 465–491.

Glover, F. (1973), 'Convexity cuts and cut search', *Operations Research* **21**, 123–134.

Glover, F., Klingman, D. & Phillips, N. V. (1992), *Network Models in Optimization and Their Applications in Practice*, John Wiley and Sons, New York.

Gomory, R. E. (1963), An algorithm for integer solutions to linear programs, *in* R. Graves & P. Wolfe, eds, 'Recent Advances in Mathematical Programming', McGraw–Hill, New York, pp. 269–302.

Guisewite, G. M. & Pardalos, P. M. (1990), 'Minimum concave–cost network flow problems: Applications, complexity, and algorithms', *Annals of Operations Research* **25**, 75–100.

Hochbaum, D. S. & Segev, A. (1989), 'Analysis of a flow problem with fixed charges', *Networks* **19**, 291–312.

Hoffman, K. R. (1981), 'A method for globally minimizing concave functions over convex sets', *Mathematical Programming* **20**, 22–32.

Horst, R. (1976), 'An algorithm for nonconvex programming problems', *Mathematical Programming* **10**, 312–321.

Horst, R. (1984), 'On the convexification of nonlinear programming problems: An applications–oriented survey', *European Journal of Operations Research* **15**, 382–392.

Horst, R. (1990), 'Deterministic methods in constrained global optimization: Some recent advances and new fields of application', *Naval Research*

Logistics **37**, 433–471.

Horst, R., Pardalos, P. M. & Thoai, N. V. (1995), *Introduction to Global Optimization*, Klewer Academic Publishers, Berlin.

Horst, R. & Tuy, H. (1996), *Global Optimization: Deterministic Approaches*, third edn, Springer-Verlag, Dordrecht.

Jacobsen, S. E. (1981), 'Convergence of a tuy–type–algorithm for concave minimisation subject to linear inequality constraints', *Applied Mathematics and Optimization* **7**, 1–9.

Jarvis, J. J., Rardin, R. L., Unger, V. E., Moore, R. W. & Schimpeler, C. C. (1978), 'Optimal design of regional wastewater systems: A fixed-charge network model', *Operations Research* **26**, 538–550.

Khan, A. M. (1987), 'Solid-waste disposal with intermediate transfer stations: An application of the fixed-charge location problem', *Journal of the Operational Research Society* **38**, 31–37.

Klincewicz, J. G. (1990), 'Solving a freight transport problem using facility location techniques', *Operations Research* **38**, 99–109.

Klingman, D., Napier, A. & Stutz, J. (1974), 'Netgen: A program for generating large scale capacitated assignment, transportation, and minimum cost network flow problems', *Management Science* **20**, 814–821.

Klingman, D., Randolph, P. H. & Fuller, S. W. (1976), 'A cotton ginning problem', *Operations Research* **24**, 700–717.

Kumar, P. C. (1988), 'A conceptual decison model for impact analysis of industry regulation: A milp approach', *Computers and Operations Research* **15**, 369–379.

Lamar, B. W. (1993*a*), 'An improved branch and bound algorithm for minimum concave cost network flow problems', *Journal of Global Optimization* **3**, 261–287.

Lamar, B. W. (1993*b*), A method for solving networkflow problems with general nonlinear arc costs, *in* D.-Z. Du & P. M. Pardalos, eds, 'Network Optimization Problems: Algorithms, Applications and Complexity', World Scientific, Singapore, pp. 147–167.

Lamar, B. W. (1995), 'Nonconvex optimization over a polytope using generalized capacity improvement', *Journal of Global Optimization* **7**, 127–142.

Lamar, B. W., Sheffi, Y. & Powell, W. B. (1990), 'A capacity improvement lower bound for fixed charge network design problems', *Operations Research* **38**, 704–710.

Lamar, B. W. & Wallace, C. A. (1997), 'Revised–modified penalties for fixed charge transportation problems', *Management Science* **43**, 1431–1436.

Land, A. H. & Doig, A. G. (1960), 'An automatic method for solving discrete programming problems', *Econometrica* **28**, 479–520.

McKeown, P. (1975), 'A vertex ranking procedure for solving the linear fixed charge problems', *Operations Research* **23**, 1182–1191.

Murty, K. G. (1969), 'Solving the fixed charge problem by ranking the extreme points', *Operations Research* **16**, 268–279.

Nemhauser, G. L. & Wolsey, L. A. (1988), *Integer and Combinatorial Optimization*, Wiley, New York.

Nemhauser, G. L. & Wolsey, L. A. (1989), Integer programming, *in* G. L. Nemhauser, A. H. G. Rinnooy Kan & M. J. Todd, eds, 'Optimization', Vol. 1 of *Handbooks in Operations Research*, Klewer Academic Publishers, Dordrecht, pp. 447–527.

Palekar, U. S., Karwan, M. H. & Zionts, S. (1990), 'A branch–and–bound method for fixed charge transportation problems', *Management Science* **36**, 1092–1105.

Pardalos, P. M. & Rosen, J. B. (1987), *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Computer Science, Springer-Verlag, Berlin.

Read, E. G. & Ring, B. (1995), *Dispatch Based Pricing*, Trans Power New Zealand, Wellington.

Rinnooy Kan, A. H. G. & Timmer, G. T. (1989), Global optimization, *in* G. L. Nemhauser, A. H. G. Rinnooy Kan & M. J. Todd, eds, 'Optimization', Vol. 1 of *Handbooks in Operations Research*, Klewer Academic Publishers, Dordrecht, pp. 631–662.

Rothfarb, B., Frank, H., Rosenbaum, D. M. & Steiglitz, K. (1970), 'Optimal design of offshore natural-gas pipeline systems', *Operations Research*

18, 992–1020.

Ryoo, H. S. & Sahinidis, N. V. (1996), 'A branch–and–reduce approach to global optimization', *Journal of Global Optimization* **8**, 107–138.

Shectman, J. P. & Sahinidis, N. V. (1998), 'A finite algorithm for global minimization of separable concave programs', *Journal of Global Optimization* **12**, 1–36.

Soland, R. M. (1974), 'Optimal facility location with concave costs', *Operation Research* **22**, 373–382.

Stroup, J. W. (1967), 'Allocation of launch vehicles to space missions: A fixed-cost transportation problem', *Operations Research* **15**, 1157–1163.

Taha, H. A. (1973), 'Concave minimisation over a convex polyhedron', *Naval Research Logistics Quarterly* **20**, 533–548.

Thakur, N. V. (1990), 'Domain contraction in nonlinear programming: Minimizing a quadratic concave function over a polyhedron', *Mathematics of Operations Research* **16**, 390–407.

Tomlin, J. A. (1971), 'An improved branch and bound method for integer programming', *Operations Research* **16**, 1070–1075.

Tuy, H. (1964), 'Concave programming under linear constraints', *Soviet Mathmatics* **5**, 1437–1440.

Tuy, H. & Horst, R. (1988), 'Convergence and restart in branch–and–bound algorithms for global optimization. application to concave minimization and d.c. problems', *Mathematical Programming* **41**, 161–183.

Vasko, F. J., Newhart, D. D., Stott, K. L. & Wolf, F. E. (1996), 'Using a facility location algorithm to determine optimum cast bloom lengths', *Journal of the Operational Research Society* **47**, 341–346.

Yaged, B. (1971), 'Minimum cost routing for static network models', *Networks* **1**, 139–172.

Zwart, P. B. (1974), 'Global maximization of a convex function with linear inequality constraints', *Operations Research* **22**, 602–609.